

Hochschule für Technik und Wirtschaft Mittweida (FH)
Fachbereich Informationstechnik / Elektrotechnik

Konzeption und Implementierung eines dynamischen Tarifrechners

Diplomarbeit
im Studiengang Multimediatechnik

Martin Patzig

Betreuer: Prof. Dr.-Ing. Frank Zimmer
Dipl. Inf. A. von Falkenburg

Inhaltsverzeichnis

1. Einleitung	7
1.1. Motivation	7
1.2. Zielstellung	8
1.3. Aufbau	8
1.4. Aufgabenstellung	9
1.4.1. Abgrenzung	9
2. PowerCommerce	11
2.1. Funktionalität	11
2.2. Aufbau	12
3. Tarife	14
3.1. Allgemeine Betrachtungen	14
3.2. SmartMetering	16
3.2.1. Allgemein	16
3.2.2. Technische Grundlagen	17
3.2.3. Entwicklungen	18
4. Konzeption	21
4.1. Ist-Analyse	21
4.1.1. Tarifmodell	21
4.1.2. Tarifrechner	26
4.1.3. Produktkonfigurator	33
4.1.4. Schnittstellen	37
4.2. Soll-Kriterien	37
4.2.1. Allgemein	38
4.2.2. Datenmodell	38
4.2.3. Tarifrechner	39
4.2.4. Produktkonfigurator	39
4.3. Konzeption	39
4.3.1. Datenmodell	39
4.3.2. Tarifrechner	43
4.3.3. Produktkonfigurator	45
5. Implementierung	49
5.1. Tarifmodell	49

Inhaltsverzeichnis

5.2. Tarifrechner	52
5.3. Produktkonfigurator	55
5.4. Schnittstellen	59
6. Ergebnis	60
6.1. Zusammenfassung	60
6.2. Ausblick / Weiterentwicklung	62
6.2.1. Produkte und Tarife - Export in das Abrechnungssystem	62
6.2.2. Verbrauchshistorie mit Preisinformationen	62
6.2.3. Zeitvariable Tarife	62
6.2.4. Liefergebietsabhängige Preise	63
6.2.5. Einmalige Boni	63
6.2.6. „Mindestens“ und „höchstens“ Filter	63
Literaturverzeichnis	65
A. Anhang	66

Abkürzungsverzeichnis

ITC Internet Trade Center AG

ITK Informations- und Telekommunikationstechnologie

EVU Energieversorgungsunternehmen

GSM Global System for Mobile Communications

Ajax Asynchronous JavaScript and XML

DOM document object model

ENSO ENSO Energie Sachsen Ost AG

AGB Allgemeine Geschäftsbedingungen

VGL Vertragsgrundlagen

DSL Digital Subscriber Line

XML Extensible Markup Language

Abbildungsverzeichnis

2.1. Aufbau PowerCommerce, Quelle: [POCO]	12
3.1. Ferraris-Stromzähler	17
3.2. Intelligenter Zähler	17
3.3. Demand-Response-Programme, Quelle: [wBMWi]	19
4.1. Derzeitiges Tarifmodell	24
4.2. Ergebnisliste Produktrechner PowerCommerce	26
4.3. Verbrauchseingabe Verivox, Quelle: [VERI]	29
4.4. Ergebnisliste TopTarif, Quelle: [ToTa]	29
4.5. Ergebnisliste ENSO, Quelle: [ENSO]	31
4.6. Ergebnisliste Energie SaarLorLux, Quelle: [SaLoLu]	32
4.7. Prozess „Produkt anlegen“ im PowerCommerce	34
4.8. Produktauswahl im Prozess „Produkt ändern“ im PowerCommerce	35
4.9. Tarifformel im Prozess „Produkt erweitern“ im PowerCommerce	36
4.10. Neues Tarifmodell	42
4.11. Aktivitätsdiagramm Tarifrechner	44
4.12. Anwendungsfalldiagramm Produktkonfigurator	47
4.13. Aktivitätsdiagramm Produkt erstellen	48
5.1. Objekte einer Tarifooption 'Preisgarantie'	51
5.2. Nutzereingaben und Tarifooptionen im Produktrechner	52
5.3. Tarifooptionen im Produktkonfigurator pflegen	55
5.4. Tarifformel im Produktkonfigurator pflegen	57

1. Einleitung

1.1. Motivation

Die Firma ITC Internet Trade Center AG entwickelt seit 1997 Softwarelösungen verstärkt für Energieversorgungsunternehmen (EVU). Da der Kundenkreis zu Beginn nur klein war, wurden nur wenige Entwickler zur Erstellung der Software benötigt. Durch den erfolgreichen Verkauf von Applikationen stieg der Bekanntheitsgrad der Firma stetig an, sodass weitere Programmierer im ITC beschäftigt werden konnten.

Derzeit arbeiten ungefähr 50 Mitarbeiter im Internet Trade Center. Für die Entwicklung kundenspezifischer Softwaresysteme programmieren drei bis fünf Angestellte in einem Team. Dabei kann ein Entwickler auch an mehreren Projekten gleichzeitig beteiligt sein.

Mit der Branchensoftware PowerCommerce wird dem EVU eine internetbasierende Kundenbetreuung geboten, die nach seinen speziellen Wünschen prozessspezifisch und optisch angepasst wird. Durch diese Anwendung werden Verwaltungsprozesse des Energieversorgers optimiert und die Kommunikation zwischen Versorger und Endverbraucher erheblich verbessert.

Immer mehr EVU bieten ihren Kunden eine Vielzahl von Tarifen an, damit dieser seinen Vertrag den individuellen Bedürfnissen anpassen kann. Es gewinnen zunehmend Tarifoptionen und Module die Oberhand, aus denen sich der Verbraucher *seinen* Tarif zusammenbauen kann. Aufgrund des Aufkommen reiner Vertriebsgesellschaften entstand eine enorme Vielfalt der Produktpaletten am Energiemarkt. Durch die gestiegene Komplexität der Tarifvielfalt ist eine Anpassung der derzeitig verwendeten Datenstruktur des PowerCommerce notwendig, um den Ansprüchen der Energieversorger gerecht zu werden. Diese Änderungen ziehen eine Reihe prozessspezifischer Neuerungen nach sich. Auf der Seite des Endkunden soll, zur Beratung bei der Tarifauswahl, ein neuer Tarifrechner entstehen. Der Energieversorger selbst wird mit einem Produktkonfigurator die Möglichkeit haben, neue Tarife zu erstellen und bestehende zu pflegen.

1.2. Zielstellung

Die vorliegende Arbeit soll eine Analyse des derzeit implementierten Tarifdatenmodells wiedergeben und Optimierungspunkte herausarbeiten. Weiterhin wird die Umsetzung der Punkte, sowie die daraus entstehenden Prozessanpassungen, beschrieben werden.

Die einfache Nutzbarkeit des Produktkonfigurators durch einen Sachbearbeiter im EVU ist ein entscheidender Faktor für die Kunden vom ITC. Deshalb sollen verschiedene Möglichkeiten der Umsetzung dieses Administrationswerkzeuges diskutiert werden.

Eine Wertschätzung der am Ende entstandenen Neuerungen soll weitere mögliche Verbesserungsansätze aufzeigen, um eine optimale Lösung der Tarifproblematik im PowerCommerce zu realisieren.

1.3. Aufbau

Einführend wird in dieser Arbeit das Kundenportal PowerCommerce allgemein beschrieben und dessen Funktionen vorgestellt. Neben einem Modulüberblick wird der prinzipielle Aufbau der Applikation skizziert.

Im anschließenden Kapitel geht der Autor auf die Entwicklung der Tarife auf dem Strommarkt ein. Die ansteigende Komplexität der Produkte und deren Ursachen sollen dabei geklärt werden. Die zukünftige Einführung von SmartMetering wird dazu führen, dass neue Tarifsysteme sich etablieren. Die Frage : „Was ist SmartMetering?“ ist ein weiterer Bestandteil des Kapitels.

Der folgende Abschnitt wird die Folgen der Tarifentwicklung für PowerCommerce beschreiben. Der aktuelle Implementierungsstand sowie die Anforderungen für ein neues System werden analysiert. Da die Tarife ein wesentlicher Bestandteil der Geschäftslogik von PowerCommerce sind, gibt es verschiedenste Punkte, die im Zuge der Neuentwicklung des Modells mit bedacht und angepasst werden sollen. Ein wesentlicher Schwerpunkt stellt dabei die Realisierung der Regionalstruktur bzw. des Liefergebietes für die Tarife dar. Nötige Anpassungen und Verbesserungsvorschläge dafür werden einen Teil des Kapitels einnehmen.

In vielen Abrechnungssystemen erfolgt eine Unterscheidung zwischen Tarifen und Produkten. Da PowerCommerce diese Differenzierung derzeit noch nicht vornimmt und um in den einzelnen Textpassagen eine mehrfache Wortverwendung zu vermeiden, werden Produkte und Tarife in dieser Arbeit als ein und dasselbe betrachtet. Im Abschnitt 6.2 wird in einem möglichen Erweiterungsschritt näher auf die Details dieser Unterscheidung eingegangen.

1.4. Aufgabenstellung

Für den Customer-Self-Service PowerCommerce soll ein dynamischer Tarifrechner konzipiert und implementiert werden. Wesentliche Anforderungen sind dabei die Berücksichtigung einer wachsenden Komplexität der Tarife, sowie die Möglichkeit der Berechnung von last- und zeitvariablen Tarifen.

Für den Kunden des Energieversorgers soll eine Tarifiermittlung aufgrund der Eingabe seiner Verbrauchsstelle erfolgen. Die Auswahl der Tarife soll dabei durch zusätzliche Filterkriterien noch weiter eingeschränkt werden können. Der Mitarbeiter des EVU soll die Möglichkeit haben, über eine Administrationsoberfläche neue Tarife zu erstellen und bestehende zu erweitern.

Der Tarifrechner soll zum einen voll in das bestehende PowerCommerce integrierbar sein, um eine Einfügung des neuen Tarifmodelles in die bereits vorhandenen Geschäftsprozesse zu ermöglichen. Zum anderen ist aber eine Nutzung als Stand-alone-Applikation wünschenswert.

Über verschiedene Schnittstellen soll es möglich sein, bestehende Tarife in den Rechner zu importieren. Diese Tarife können beispielsweise aus einem Abrechnungssystem oder einer älteren Version von PowerCommerce stammen.

1.4.1. Abgrenzung

In Folge einer weiteren Diplomarbeit beim ITC findet derzeit eine Umstellung des Frameworks von PowerCommerce statt. Inhalt der Arbeit war ein Vergleich verschiedenster Basistechnologien zur Realisierung von Geschäftsprozessen. Dabei stand eine schnelle, unkomplizierte und einfach wartbare Anpassung von den im PowerCommerce vorhandenen Geschäftsabläufen im Vordergrund.

Als neues Framework hat man sich aufgrund der Diplomarbeit für die Webflow-Technologie des Spring-Frameworks entschieden. Damit können Prozessschritte ohne großen Programmieraufwand umsortiert werden und ein Austausch von Geschäftsrou-tinen durch andere ist problemlos möglich. Ein weiterer Vorteil der Neuerung ist, dass einzelne Abläufe nicht nur innerhalb der Applikation laufen können, sondern auch als Webservice separat zur Verfügung stehen.

Dieser Systemwechsel zieht bei einer mit der Zeit so enorm gewachsenen Anwendung wie PowerCommerce eine Vielzahl von Neuimplementierungen und Anpassungen nach sich. In dem Rahmen werden natürlich auch grundlegende Dinge neu überarbeitet, die im bestehenden System immer wieder als unperformant, verbesserungswürdig oder sicherheitsrelevant betrachtet wurden. Historisch gewachsene Fehler oder Ungenauigkeiten

1. Einleitung

sollen so von Anfang an vermieden werden.

Damit eine Anpassung der bisherigen Implementierung von PowerCommerce bezüglich der neuen Tariflandschaft vermieden werden kann, sollen die hier gewonnenen Erkenntnisse und Änderungen ausschließlich durch das neue Framework umgesetzt werden. Durch den enormen Arbeitsaufwand der Neuentwicklung von PowerCommerce und die Vielzahl der parallel dazu anfallenden Projekte im ITC ist es während des Entstehens dieser Diplomarbeit jedoch nicht möglich gewesen, ein umfassendes Basissystem zu schaffen, welches eine Umsetzung der Tarifanpassungen ermöglicht. Die in dieser Arbeit erarbeitete Konzeption bezieht sich dabei auf den vollen Umfang der geplanten Änderungen. In der Implementierung werden aber nur die Änderungen behandelt, welche derzeit tatsächlich umgesetzt werden konnten.

Im folgenden soll beschrieben werden von welchen geplanten Neuerungen derzeit noch abgesehen werden muss und welche aufgrund ihrer Dringlichkeit in das bereits bestehende System implementiert werden.

Das Datenmodell zu den neuen Tarifen kann aufgrund der Fülle der Änderungen nicht komplett umgesetzt werden. Da jedoch viele EVUs die Realisierung von Tarifoptionen schnellstmöglich benötigen, werden diese Änderungen vorgenommen.

Der bestehende Tarifrechner wird um die Tarifoptionen erweitert. Dafür werden sowohl die Eingabe der Nutzerangaben als auch die eigentliche Berechnung des Tarifes erweitert.

Weiterhin wird der Produktkonfigurator überarbeitet, um Tarifoptionen pflegen zu können. Auf die Trennung der einzelnen Prozessschritte in neue, eigenständige Geschäftsabläufe wird aufgrund der Fülle der Änderungen zurzeit verzichtet.

Die im Rahmen dieser Diplomarbeit nicht realisierten Punkte werden bei der Weiterentwicklung von der nächsten PowerCommerce-Version mit umgesetzt.

2. PowerCommerce

PowerCommerce ist eine Webanwendung zur Automatisierung und Optimierung von Geschäftsprozessen der Energieversorger. Durch diese E-Business-Applikation haben sowohl Privatkunden als auch Geschäftskunden die Möglichkeit Versorgerverträge abzuschließen oder zu verwalten. Diese Mehrwertdienste erweitern die Serviceoptionen des Versorgungsunternehmens und beeinflussen die Beziehung zum Kunden positiv. Durch den Einsatz von PowerCommerce können EVU unter anderem Verwaltungs- und Bearbeitungskosten für die Zählerstandserfassungen einsparen. Diese werden nicht per Post, sondern direkt über den Online-Service abgewickelt.

Der modulare Aufbau von PowerCommerce bietet viele Änderungsmöglichkeiten um das Online-Portal den Wünschen des Versorgers anzupassen. So können vertraute Geschäftsgepflogenheiten problemlos in die Internetkundenbetreuung übernommen werden.

2.1. Funktionalität

Die Geschäftsprozesse von PowerCommerce lassen sich im Allgemeinen in fünf Module aufteilen. Für einen Großteil der Prozesse benötigt der Kunde einen Login, damit seine persönlichen Vertragsdaten geladen werden können. Dies setzt eine einmalige Registrierung am System voraus. Anwendern ohne Zugangsberechtigung stehen nur Informationsprozesse zur Verfügung.

Modulüberblick

Basisdienste Dem angemeldeten Nutzer werden Prozesse zur Änderung seiner Stammdaten, Logindaten sowie Kontaktdaten geboten. Außerdem hat er hier die Möglichkeit weitere Verträge oder Kundennummern für die Verwaltung in seinem Nutzerprofil zu registrieren.

Produkte Dieses Modul stellt dem Kunden die vom Energieversorgungsunternehmen angebotenen Tarife für die Sparten Strom, Gas, Wasser und Wärme (je nach EVU) vor. Durch die Eingabe einer Postleitzahl kann die Verfügbarkeit einzelner Produk-

te geprüft werden. Der Tarifrechner findet anhand von Benutzerangaben, wie z.B. Durchschnittsverbrauch, den für den Kunden günstigsten Tarif. Ist der Verbrauch dem Kunden nicht bekannt so kann dieser mit Hilfe des Gerätrechners simuliert werden. Diese Informationsprozesse sind sowohl angemeldeten als auch nicht angemeldeten Nutzer zugänglich.

Abrechnung Dieser Abschnitt deckt alle Geschäftsprozesse ab, die zur finanziellen Abwicklung des Kundenvertrages beitragen. Dazu zählen die Bankdatenänderung, Abschlagsänderung und Zählerstandserfassung. Die Verbrauchshistorie liefert eine grafische Auswertung der Verbrauchsdaten eines definierten Zeitraums.

Service Kundenanfragen sowie Störungsmeldungen können über dieses Modul realisiert werden.

2.2. Aufbau

Der Aufbau von PowerCommerce beschreibt ein 3-Schichtmodell. Dazu werden die Komponenten in die Präsentationsschicht, eine Geschäftsebene und die Datenhaltungsebene eingeteilt.

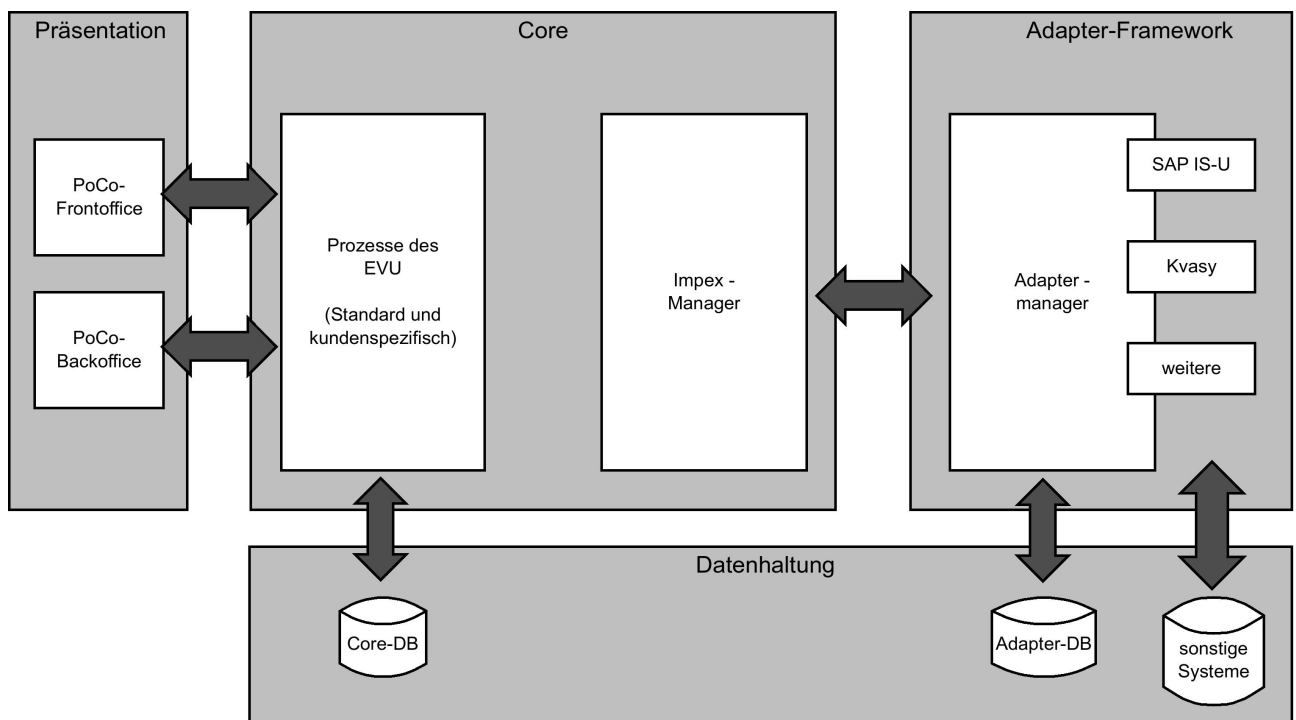


Abbildung 2.1.: Aufbau PowerCommerce, Quelle: [POCO]

2. PowerCommerce

Durch die Mandantenfähigkeit von PowerCommerce ist es möglich eine Applikation für mehrere Energieversorgungsunternehmen zu nutzen. Dabei können Geschäftsprozesse gemeinschaftlich genutzt werden, aber auch individuell für einen Mandanten angepasst werden. Die Präsentation der Anwendung erfolgt für jeden Mandanten separat.

Jedem Mandanten stehen dabei zwei mögliche Zugangssvarianten des Online-Kundenportals zur Verfügung. Die Energieabnehmer werden dabei den Einstieg über das Front-Office nutzen. Dieses enthält die für den Kunden nützlichen Prozesse, um Vertragsdaten zu bearbeiten oder mit dem Energieversorger in Kontakt zu treten.

Die Sachbearbeiter des Energieversorgers wählen in der Regel den Zugang über das Back-Office¹. Darüber ist eine Nutzung spezieller Sachbearbeiterprozesse, wie Transaktionsliste oder Nutzerverwaltung, möglich.

Das abstrakte Framework von PowerCommerce ist in einem Core zusammengefasst. Das Referenzprojekt beinhaltet eine Reihe von Standardimplementierungen für die benötigten Kundenprozesse. Auch diese befinden sich in der Core-Implementierung. Abweichungen davon werden für jeden Kunden speziell durchgeführt und in die Geschäftslogik eingebunden.

Die Kundendaten bezieht PowerCommerce aus den Abrechnungssystemen der Energieversorgungsunternehmen. Da es eine Vielzahl solcher Systeme gibt, wurde ein Adapterframework entwickelt. Dieses beinhaltet eine Sammlung von Adaptern zu verschiedenen Systemen. Um dieses unabhängig von PowerCommerce zu betreiben, erfolgte eine Entwicklung als eigenständige Adapterapplikation. Der Datenabgleich zwischen Geschäftslogikdaten und den Adapterdaten wird über ein Import- und Exportmanagement realisiert.

¹PowerCommerce verfügt über zwei separate Zugangsbereiche. Das „Front-Office“ enthält die für den Kunden zugänglichen Prozesse, wie Neuanmeldung oder Bankdatenänderung. Der „Back-Office“-bereich ist nur für Sachbearbeiter konzipiert und stellt administrative Prozesse dar. Die Trennung der beiden Bereiche erfolgt über ein Rechte- und Rollenmanagement.

3. Tarife

3.1. Allgemeine Betrachtungen

Ein Tarif beschreibt im Allgemeinen eine Sammlung von Bedingungen bzw. Preisen für bestimmte Leistungen. Diese umfassen beispielsweise eine geltende Vertragslaufzeit oder eine zu beachtende Kündigungsfrist.

Tarife findet man in allen erdenklichen Bereichen der Wirtschaft und des täglichen Lebens:

Arbeitstarif bzw. Tarifverträge regelt Bedingungen zum Verhältnis zwischen Tarifvertragsparteien am Arbeitsplatz.

Tarife für Telekommunikationsmedien regeln Preise und Leistungsmerkmale zur Nutzung des entsprechenden Mediums.

...

Durch eine Variierung der Werte der geltenden Tarifbedingungen regulieren die Tarifanbieter die Preise. Diese Festlegungen haben eine Anzahl von verschiedenen gängigen Konstellationen herausgebildet. Vor allem in der Telekommunikationsbranche haben sich unterschiedlichste Modelle herausgearbeitet:

Leistungsbezogene Preisabrechnung Die einfachste Form der Preisbildung. Es gibt hierbei meist einen Grundpreis für die Bereitstellung der Leitung und einen Arbeitspreis pro verwendeter Leistungseinheit. Beispielsweise berechnet ein Anbieter eine Grundgebühr von 12 Euro und erhebt eine Nutzungsgebühr von 5 Eurocent pro Nutzeinheit. Der Endpreis ergibt sich dabei aus:

$$\text{Preis} = \text{Grundgebühr} + \text{Nutzungsgebühr} * \text{Verbrauchseinheit}$$

Volumentarife Bei dieser Form des Tarifes entrichtet der Kunde einen Betrag für ein festes Nutzungskontingent pauschal. Bleibt er mit seiner Nutzung innerhalb dieses Rahmens, fallen für ihn keine weiteren Kosten an. Überschreitet er aber die vorgegebenen Nutzungsgrenzen berechnet der Anbieter der Dienstleistung eine Gebühr nach der leistungsbezogenen Abrechnung (s.o.).

3. Tarife

$$\text{Preis} = \text{Pauschalgebühr} + \text{Zusatzkosten},$$

wenn Nutzung > Nutzobergrenze: Zusatzkosten = Nutzungsgebühr * Einheit

wenn Nutzung ≤ Nutzobergrenze: Zusatzkosten = 0

Pauschaltarife / Flatratetarife Als Pauschaltarife bezeichnet man die Entrichtung eines fixen Betrages bei unbegrenzter Nutzungsmöglichkeit. Diese Tarifform ist die derzeit am häufigsten beworbene und auch meist genutzte im Telekommunikationssektor. Dabei können Kunden gegen eine festgeschriebene Grundgebühr beispielsweise ihren Telefonanschluss so oft und so lange nutzen wie sie möchten.

$$\text{Preis} = \text{fixer Grundpreis}$$

Um sich weiterhin von der Konkurrenz abzuheben und für den Kunden attraktive und individuelle Preise zu erstellen, bedienen sich die Dienstleistungsanbieter weiterer Variationsmöglichkeiten. In vielen Branchen führen innovative Ansätze und eine intelligente Preisgestaltung zu steigenden Umsätzen.¹

Die Wahl einzelner Vertragsbedingungen wird dazu an bestimmte Preise gekoppelt. Somit ist eine Preisreduzierung denkbar, wenn der Kunde sich für eine längere Laufzeit des Vertrages entschließt, ein Lastschriftverfahren zur Begleichung seiner Rechnungen wählt oder einen Service über kostenpflichtige Kommunikationswege wählt. Gegenteilig wirken sich Überweisungszahlungen, persönliche Kundenberater und gedruckte, postalisch versandte Rechnungen in steigenden Preisen aus.

Durch die Instrumentalisierung der einzelnen Tarifbedingungen als variable, preisbeeinflussende Komponenten, verlagert man die Preisfestlegung weg vom Dienstleistungsanbieter und hin zum Kunden. Der Preis wird dem Kunden nicht mehr aufdiktiert, sondern er kann innerhalb der Grenzen des Anbieters, frei entscheiden, was sein Vertrag beinhaltet und letztendlich wie sich der Preis entwickelt.

Am Energiemarkt prägte bisweilen eine einseitige Tariflandschaft das Bild, innovative Preismodelle waren eher selten zu finden². Das leistungsbezogene Tarifmodell war die üblichste Form der vertraglichen Beziehung zwischen Anbietern und Kunden. Die Liberalisierung des Strommarktes 1999 sowie der Wegfall der Bundestarifordnung Elektrizität (BTOElt)³ brachte hier eine deutliche Dynamik. Diese Liberalisierung erlebte ab 2004 auch der Gasmarkt in Deutschland. Um sich von der schnell wachsenden Anzahl an Konkurrenten abzuheben, lockten die Energieversorger ihre Kunden mit Wechselprämien und Bonizahlungen. Die individuelle Gestaltung der Tarifbestandteile und des daraus resultierenden Preises findet in den Produktkonfiguratoren einiger EVU Anwendung. Da-

¹[EM0408]

²[EM0408]

³<http://beck-online.beck.de/default.aspx?bcid=Y-100-G-BTOElt>

3. Tarife

bei können Zahlungsmethode und Vertragslaufzeit, Serviceleistungen und Rechnungsart eine Auswirkung auf den Preis haben.

Sehr deutlich ist diese Art der Preisgestaltung beim CleverClicker der Firma Clevergy GmbH & Co. KG ausgeprägt⁴. Durch eine Slotmaschine kann der Kunde sich sein Produkt in den Kategorien Energiemix, Kautions, Preisgarantie, Vertragslaufzeit, Kundenservice, Rechnung, Zahlungsweise und Zahlungszeitpunkt selbst gestalten. Jeder der gewählten Parameter hat dabei einen direkten Einfluss auf den Vertragspreis.

Das Modell der Volumentarife findet sich auch hier an mancher Stelle wieder. Einige EVU bieten ihren Kunden beispielsweise den Erwerb von Strompaketen an. Er kann Strompakete zu 500kWh mit einem höheren Arbeitspreis beziehen und damit den Erwerb von regenerativen Energiegewinnungseinrichtungen fördern⁵.

Eine weitere Entwicklung der Tariflandschaft wird SmartMetering hervorrufen. Durch die aktuellen Ablesewerte sind zeit- und lastvariable Tarife möglich.

3.2. SmartMetering

3.2.1. Allgemein

“SmartMetering” kann mit intelligenter Zählwerterfassung übersetzt werden. Die zunehmende Notwendigkeit des rationalisierten Energieeinsatzes, um klimatische Veränderungen vorzubeugen, und die rasanten Entwicklungen in der ITK bringen neue Nutzungsmöglichkeiten für die Energieerzeugung, -verteilung und -versorgung mit sich. Eine Kommunikation zwischen dem EVU und den Zählgeräten beim Kunden wird durch eine immer weiter verbreitete Anbindung der Bevölkerung an das Breitbandnetz, sowie die nahezu flächendeckende Verfügbarkeit von drahtlosen Kommunikationsmedien, wie GSM, möglich.⁶ Durch diese Verbindung werden neue Verfahren und Dienstleistungen im Bereich der Heimautomatisierung seitens der Energieversorger realisierbar.

Im Haushaltsbereich sind die Potenziale zum einen in der Visualisierung des Energieverbrauches und einer resultierenden Einsparung, aber auch in der Einführung von zeitvariablen Tarifen⁷ zu finden. Der Einsatz von intelligenten Zählersystemen führt zu einer Preissensibilisierung des Kunden und kann Geschäftsprozesse, wie Lieferantenwechsel oder das Messen und Abrechnen von Verbrauchsdaten automatisieren.

⁴<http://www.clevergy.de>

⁵<http://www.sws.speyer.de>

⁶[wBMWi]

⁷Durch intelligente Stromzähler ist es möglich, in einem Messgerät mehrere Zeitscheiben zu definieren. Diese messen anschließend den Verbrauch nur in dem zugewiesenen Zeitraum. Durch dieses Verfahren können Tarife erstellt werden, die zeitabhängig ihre Preise verändern.

3.2.2. Technische Grundlagen

Ein SmartMeter, intelligenter Zähler (Abbildung 3.2 Seite 17), ist ein, mit zusätzlichen Funktionen, zur Datenverarbeitung und Datenkommunikation, versehener Zähler. Das elektro-mechanische Zählwerk der Ferraris-Zähler (Abbildung 3.1 Seite 17) wird dabei durch ein elektronisches ersetzt. Die gemessenen Werte werden elektronisch gespeichert und können über Kommunikationsschnittstellen oder integrierte Kommunikationssysteme ausgelesen werden. Im Gegensatz zu den üblichen Ferraris-Zählern machen sie eine Jahresablesung des Stromverbrauches überflüssig.



Abbildung 3.1.: Ferraris-Stromzähler Abbildung 3.2.: Intelligenter Zähler

Bei der Zählerfernauslese unterscheidet man zwischen AMR (advanced meter reading) und AMI (advanced meter infrastructure) bzw. AMM (advanced meter management).

Beim AMR werden die Ablesewerte mittels Impulsausgang oder externer Ableseeinrichtung erfasst und per Funk oder von einem Mitarbeiter des Versorgungsunternehmens ausgelesen. Diese Ablesung erfolgt vor Ort oder von der Straße aus. Ein direkter Kontakt mit dem Zähler ist dabei nicht notwendig. AMR-Systeme beschleunigen lediglich den Vorgang des Ablesens.

Bei den AMI bzw. AMM-Systemen ist eine bidirektionale Kommunikation mit dem Zähler möglich. Diese erfolgt über ein integriertes oder separates Kommunikationssystem. Meist sind die Zähler mit einem Tarifregister ausgestattet, das eine Speicherung von bis zu viertelstundengenauen Werten ermöglicht. Die einzelnen Hersteller von AMM-Zählern bieten ihren Kunden dabei verschiedenste Funktionen (siehe [wBMWi] Seite 111). Derzeit bieten die Zählerhersteller unterschiedlichste Varianten der Realisierung des SmartMetering an.

3. Tarife

1. Der Zähler wird an den DSL-Anschluss des Verbrauchers angeschlossen, um die gemessenen Daten an den Versorger zu übermitteln. Über eine spezielle Software hat der Kunde die Möglichkeit, seinen Momentanverbrauch einzusehen. So kann er beim Ein- und Ausschalten von Geräten erkennen, welchen Einfluss das Gerät auf den Momentanverbrauch hat.
Anwendung findet dieses Szenario bei der Firma Yello Strom.⁸ Nachteil dieser Variante: Der Zugriff auf den Zähler ist nur lesend möglich.
2. Es erfolgt eine Sammlung und Aufbereitung der Zählerdaten über eine Software des Zählerherstellers. Das Angebot der Visualisierung, Verbrauchsanalyse und Kommunikation mit dem Zähler wird vom Zählerhersteller vorgegeben. Drittanwendungen haben keine Möglichkeit zusätzliche Informationen zu gewinnen oder Funktionen des Zähler zu beeinflussen.
3. Der Zählerhersteller realisiert die Sammlung der Daten und stellt eine performante Schnittstelle zur Gewinnung der Informationen für Drittanwendungen bereit.

3.2.3. Entwicklungen

Durch die aktuellen Daten hat der Kunde ständig die Möglichkeit sich über seinen Momentanverbrauch zu informieren. Diese Sensibilisierung des Kunden führt zur Optimierung des Nutzungsverhaltens und zur Senkung der Kosten. Nach einer Studie von A. T. Kearney sind demnach Einsparungen zwischen fünf und zehn Prozent zu erwarten⁹. Da weniger Energie verbraucht wird, kann der CO_2 -Ausstoß erheblich verringert werden. Der Einsatz dieser Zähler eröffnen neue Geschäftsmodelle für die EVU:

- monatliche/ vierteljährliche/ halbjährliche, verbrauchsgenaue Abrechnung
- Anzeige des Momentanverbrauches im Haushalt
- Verbrauchsanalyse
- Erkennen von Verbrauchsspitzen
- Alarm bei Überschreitung definierter Schwellwerte
- zeit- und lastvariable Tarife

Die Stromnachfrage in den einzelnen Haushalten ist in ihrem Verlauf ziemlich genau vorhersehbar. Sie wird dabei nur von externen Einflüssen, wie Wetter- und Lichtverhältnissen im Tagesverlauf beeinflusst. Das wiederkehrende Nachfragemuster erleichtert

⁸[YELLO]

⁹[http : //www.atkearney.de/content/veroeffentlichungen/pressemitteilungen_detail.php/id/50438](http://www.atkearney.de/content/veroeffentlichungen/pressemitteilungen_detail.php/id/50438)

zum einen den Planungsprozess für die Erzeuger, zum anderen erzeugt eine Abweichung von diesem Muster das Problem der Vorhaltung der Kapazitäten¹⁰. Daher liegt auch ein gesteigertes Interesse an den genauen Verbrauchsinformationen bei den Kraftwerks- und Netzbetreibern. Diese können auf die benötigte Last zeitnah reagieren und so netzgefährdende Situationen abwenden. Dabei kann eine ausgefeilte Nachfrageverwaltung (Demand-Side Management - DSM) eine große Rolle spielen.

Ein DSM beschreibt in diesem Falle alle direkten oder indirekten Maßnahmen des Energieversorgers oder Netzbetreibers, die zur Verschiebung oder Veränderung der Lasten führen. Eine spezielle Ausgestaltung dieser Maßnahmen sind die sogenannten Demand-Response-Programme. Dabei unterscheidet man zwischen preisbasierenden und anreizbasierenden Programmen.

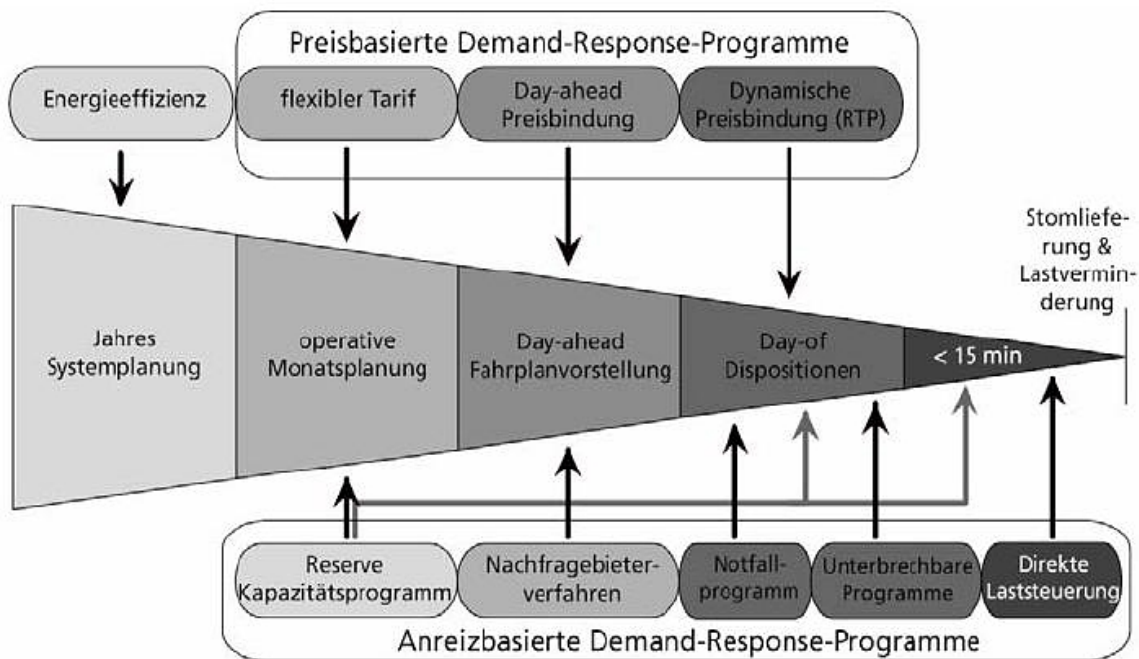


Abbildung 3.3.: Demand-Response-Programme, Quelle: [wBMWi]

Preisbasierende Programme sind zum Beispiel zeitvariable Tarife, dynamische Tarife und Spitzenlasttarife.

Zeitvariable Tarife haben festgelegte Preisstufen und geltende Intervalle. Die Zeiträume können Tageszeiten, Wochentage oder Jahreszeiten widerspiegeln. Die Preisbildung erfolgt bei zeitvariablen Tarifen anhand der durchschnittlichen Erzeugungs- und Netzkosten für den definierten Zeitraum.

¹⁰[wBMWi] Seite 73

3. Tarife

Bei dynamischen Tarifen erfolgt die Preisbildung durch externe Einflüsse der Versorgungssituation (z.B. Börsenpreise) oder ähnliche zeitnahe Preissignale. Die Vorhersage erfolgt für den der Nachfrage voranstehenden Zeitraum (z.B. stündlich, täglich).

Eine Mischform der vorangegangenen Tariformen zeigt sich in den Spitzenlasttarifen. Dabei erfolgt die Preisfestsetzung für definierte Zeiträume. Durch bestimmte (festgelegte) Situationen kann aber eine kurzfristige Änderung des Preises erfolgen.

Durch die Weitergabe der Preissignale an den Kunden kann eine elastische Stromverbrauchskurve entstehen. Der Kunde kann zu jeder Zeit selbst entscheiden, wann und zu welchen preislichen Bedingungen er das Produkt Strom einkauft.

Durch den Einsatz von Maßnahmen zur Regulierung der Nachfrage können Lastspitzen geglättet und so der Einsatz von teuren Spitzenlastkraftwerken vermieden werden. Dadurch ergeben sich geringere Stromhandelspreise für den gesamten Markt. Eine weitere Ersparnis kann der Endkunde erreichen, indem er seine Nachfrage in tariflich günstigere Perioden verlagert. Neben dem finanziellen Nutzen, steigert das veränderte Nachfrageverhalten die Versorgungszuverlässigkeit, da mit einer gleichmäßigeren Netzlast die Wahrscheinlichkeit von Ausfällen abnimmt.

4. Konzeption

4.1. Ist-Analyse

4.1.1. Tarifmodell

In der aktuellen Version von PowerCommerce ist die Abbildung von Tarifen in acht Tabellen gegliedert (Abbildung 4.1 auf Seite 24).

Allgemeine Daten, die einmalig für einen Tarif bestimmt werden, sind in der Tabelle PC_Tariff zusammengefasst. Sprachabhängige Daten werden in PowerCommerce in I18N-Tabellen¹ verwaltet. So erhält jeder Tarif übersetzte Angaben in der Tabelle PC_TariffI18N.

Tarife können verschiedene Attribute haben. Diese werden als Schlüsselwort-Wert-Paare in der Tabelle PC_TariffAttribute gespeichert.

Die Formel zur Berechnung des Preises wird in der Tabelle PC_TariffFormula definiert. Preisbestimmende Tarifbestandteile sind als Tariffparts im PowerCommerce gekennzeichnet. Sie werden in den Tabellen PC_TariffPart und PC_TariffPartI18N organisiert. Die sprachabhängigen Tarifbestandteile können, wie die Tarife an sich, verschiedene Attribute (PC_TariffPartAttribute) haben. Die Preise für die einzelnen Tarifbestandteile werden in der Tabelle PC_Price gehalten.

Die bisher beschriebenen Tabellen stellen jeweils 1:n bzw. n:1 Beziehungen zueinander her. Dabei ist zu bemerken, dass Relationen in PowerCommerce nicht auf Datenbankebene über Integritätsbedingungen² gelöst werden. Die Beziehungen der einzelnen Tabellen untereinander werden in der Geschäftslogik realisiert. Dadurch ist PowerCommerce flexibler bei der Anpassung an die einzelnen Geschäftsgepflogenheiten seiner Kunden.

Sonderfälle im Tarifmodell bilden zum einen die Tabelle PC_TariffAG und zum anderen die Tabelle PC_PostalcodeAggregation. Diese realisieren eine m:n Beziehung zwischen den einzelnen Tarifen und den Nutzergruppen bzw. den im System hinterlegten

¹Internationalisierung (engl. internationalization) wird im englischen Sprachraum gerne mit I18N abgekürzt (im engl. Wort befinden sich 18 Buchstaben zwischen I und N).

²Als Integritätsbedingungen beschreibt man die Festlegung von Schlüssel- und Fremdschlüsselbeziehungen in Datenbanksystemen. Dabei wird definiert, dass eine Fremdschlüsselspalte der einen Tabelle nur Schlüsselwerte aus dem Wertebereich einer anderen Tabelle annehmen darf.

Postleitzahlen.

Im Allgemeinen bestehen die Tabellen von PowerCommerce aus einem systemweit eindeutigen Identifikationsschlüssel ("pcID") , einem Erstellungs- und Bearbeitungsdatum ("creationDate, lastModifiedDate") und einem Statusfeld ("status"). In Abbildung 4.1 auf Seite 24 wurde auf die Darstellung dieser Felder, bis auf die pcID, verzichtet, da sie für die Veranschaulichung des Tarifmodells keinen Einfluss haben. Lediglich der Identifikationsschlüssel ("pcID") wird aufgezeigt, da über dieses Feld die möglichen Beziehungen hergestellt werden.

PowerCommerce unterstützt im Einsatz viele verschiedene Datenbanksysteme. So sind derzeit Installationen mit Oracle, MySQL und Informix stark verbreitet. Die in Abbildung 4.1 (Seite 24) angegebenen Datentypen können vom einzelnen Datenbanksystem abweichen. In der Darstellung wurde eine Angabe mit allgemein bekannten Datentypen vorgenommen.

Anbei ist eine Beschreibungsliste der einzelnen Felder der jeweiligen Tabellen:

PC_Tariff Jeder Tarif kann über einen systeminternen Namen verfügen ("name"). Meist werden aber die internationalisierten Namen dem Nutzer von PowerCommerce kommuniziert. Weiterhin existiert für jeden Tarif eine bestimmte Laufzeit ("contractTerm"), eine definierte Kündigungsfrist (cancelationperiod) und eine vorgeschriebene Vertragsverlängerungszeit ("termextension"). Die Einheiten der angegebenen Zeiten werden in Spalten mit dem Suffix "unit" gespeichert. Eine optische Darstellung des Tarifes wird über ein Bild ("image") oder eine spezielle Webseitenvorlage ("templatepath") realisiert. Diese Spalten enthalten jeweils eine Pfadangabe zu einer Datei des eigenen Servers oder einer externen Adresse. Ob beim Vertragsabschluss eine Unterschrift benötigt wird, gibt die Spalte "signatureRequired" wieder. Die Spalte zur Tarifart ("tariffType") speichert die Information, ob ein Tarif Eintarif- oder Zweitarifzähler³ unterstützt. Die dabei möglichen Zählernamen werden in der Spalte "meterNames" abgelegt. Einen Großteil der Spalten bestimmt das Liefergebiet des Tarifes. Dabei können Objekte wie Landkreise, Postleitzahlen oder Städte ein- oder ausgeschlossen werden. Das Liefergebiet umfasst die Spalten "states", "counties", "excludeCounties", "postalcodes", "excludePostalcodes", "cities", "excludeCities".

PC_TariffI18N In der Internationalisierungstabelle für den Tarif, kann der anzuzeigende Name ("displayName") und beschreibende Texte ("description", "shortDescription") für eine bestimmte Sprache ("locale") hinterlegt werden. Weiterhin können Pfade zu AGB- und VGL-Dateien eingetragen werden.

³Eintarifzähler messen den gesamten Verbrauch auf ein Zählwerk. Damit ist nur ein konstanter Arbeitspreis möglich. Zweitarifzähler unterscheiden zwischen Tagverbrauch (6Uhr - 22Uhr) und Nachtverbrauch (22Uhr bis 6Uhr). Beide Verbrauchswerte können zu unterschiedlichen Arbeitspreisen abgerechnet werden.

- PC_TariffAttribute** In dieser Tabelle können eine Vielzahl geltender Tarifattribute angelegt werden. Dabei wird dem Attributwert („attributeValue“) ein bestimmtes Schlüsselwort („attributeKey“) zugeordnet. So wird zum Beispiel derzeit die Information, ob ein Vertrag eine Unterschrift für eine Kündigung benötigt, als Tarifattribut gespeichert.
- PC_TariffFormula** Die Formel zur Preisbestimmung des Tarifes wird als Zeichenkette in der Spalte „formula“ gespeichert. Zusätzlich dazu wird beim Anlegen einer Tarifformel aus dieser Zeichenkette ein Programmobjekt generiert. Damit soll die spätere Berechnung des Tarifpreises beschleunigt werden. Das Programmobjekt der Formel wird im Feld „byteCode“ abgelegt.
- PC_TariffPart** Die Berechnung des Tarifpreises erfolgt über die mathematische Verknüpfung einzelner Tarifbestandteile. Diese werden in der Tabelle PC_TariffPart gespeichert. Die Bestandteile können zum einen fixe Preise repräsentieren (z.B. Grundpreis oder Arbeitspreis) oder zum anderen Nutzereingaben („userValueFlag“) sein. Ein Tarifbestandteil wird weiterhin über eine Größeneinheit („quantityUnit“) und einen mögliche Mengenschritt („quantityStep“) bestimmt. Der Mengenschritt ist dabei der Abstand zwischen zwei benachbarten Mengenwerten.
- PC_TariffPartI18N** Die Übersetzungen für den Namen („displayName“) und die (Kurz-) Beschreibung („description“, „shortDescription“) des Tarifbestandteils wird sprachenabhängig („locale“) in dieser Tabelle hinterlegt.
- PC_TariffPartAt** Jede Übersetzung eines Tarifbestandteiles kann mehrere Attribute haben. Für diese wird jeweils ein Anzeigenamen („displayName“) und eine Kombination aus Schlüsselwort („attName“) und Attributswert („attValue“) gespeichert.
- PC_Price** Den Tarifbestandteilen sind einzelne Preise („price“) für unterschiedliche Währungen („currencyId“) zugeordnet.
- PC_TariffAg** Mit Hilfe dieser Tabelle können einzelne Tarife den existierenden Nutzer- und Kundengruppen separat zugeordnet werden. Die Tabelle ist dabei das Verbindungsglied, um eine m:n-Beziehung zu realisieren.
- PC_PostalcodeAggregation** Ähnlich wie die PC_TariffAg dient diese Tabelle zur Realisierung einer m:n-Beziehung zwischen den im System hinterlegten Postleitzahlen und den einzelnen Tarifen. Die Angaben zum Liefergebiet in der Tabelle PC_Tariff werden beim Anlegen eines Tarifes in die einzelnen Postleitzahlen übersetzt. In der Tabelle PC_PostalcodeAggregation erfolgt dann eine genaue Zuordnung aller gültigen Postleitzahlen für das vorhandene Liefergebiet. Die Angaben in der PC_Tariff werden danach nur zur Auswertung und Anzeige im Produktkonfigurator weiterverwendet.

Bei der Analyse der derzeitigen Datenmodells fallen einige Schwachstellen auf.

4. Konzeption

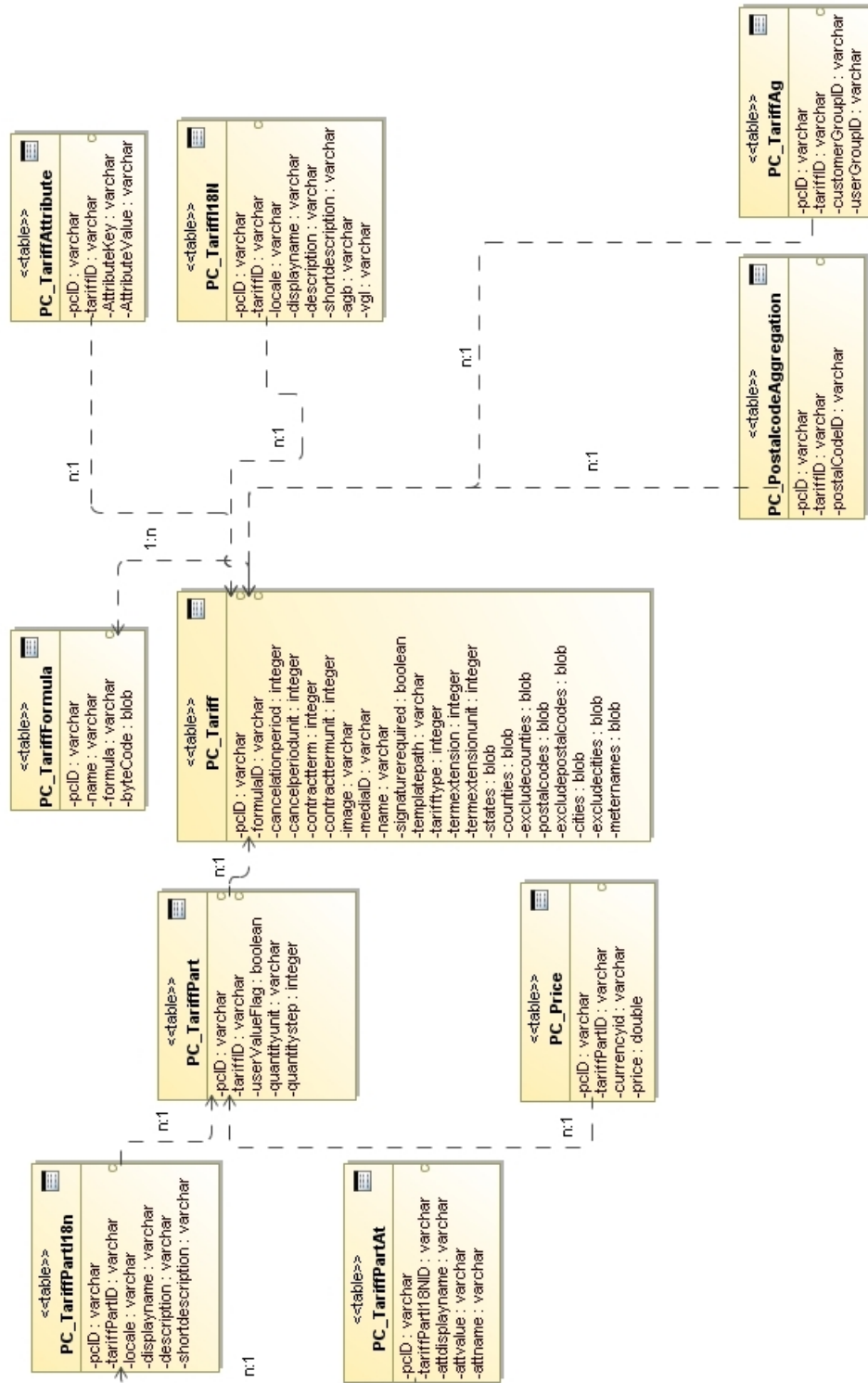


Abbildung 4.1.: Derzeitiges Tarifmodell

4. Konzeption

Da die Tarifbestandteile (*TariffPart*) durch eine Fremdschlüsselbeziehung direkt an einem bestimmten Tarif hängen, können diese nicht global gepflegt werden. Wenn also ein Tarifbestandteil bei mehreren Tarifen existiert, muss dieser auch im Datenmodell mehrfach angelegt werden.

Für Tarifbestandteile können nur zwei verschiedene Typen definiert werden: Nutzereingaben (*uservalue: true*) und Preisinformationen (*uservalue: false*). Dies stellt einen weiteren Nachteil der Datenstruktur dar.

Weiterhin gestaltet sich die Liefergebietspflege bisher sehr schwierig und aufwendig. Beim Anlegen eines Tarifes muss das gesamte Liefergebiet neu definiert werden. Die Kombination aus eingeschlossenen und ausgeschlossenen Liefergebietselementen (Landkreise, Postleitzahlen, Orte) wird dann direkt an dem Tarif in den Spalten *counties*, *excludecounties*, *postalcodes*, *excludepostalcodes*, *cities* und *excludecities* gespeichert. Eine Suche nach bestimmten Elementen des Liefergebietes mit Hilfe einer Datenbankabfrage ist aufgrund des Datentypes auch nicht möglich, sodass dafür iterativ sämtliche Datensätze durchlaufen werden müssen.

Weiterer Verbesserungsbedarf besteht für die Tabelle zur Speicherung der Tarifformel. In der bisherigen Lösung wird aus dem Formeltext eine Klasse generiert, welche anschließend in der Spalte *byteCode* abgespeichert und für alle weiteren Berechnungen herangezogen wird. Wenn eine Formel angepasst werden muss, so kann man dazu nicht nur den Formeltext in der Datenbank editieren. Es müsste weiterhin der Klassenbytecode neu generiert werden, jedoch ist diese Funktion im PowerCommerce nicht vorhanden. Ein Debugging der generierten Klassen ist auch nicht möglich, was eine Fehlersuche während der Entwicklung schwierig macht.

Für eine Neuentwicklung des Modells sollten die Tarifbestandteile global pflegbar sein und mehrere Typen zulassen können. Die Tarifformeln sollten bei Bedarf auch über die Datenbank editierbar sein. Weiterhin sollte die Berechnung der Tarifpreise transparent und nachvollziehbar erfolgen, um eine Fehlersuche zu erleichtern. Liefergebiete müssten global für alle Tarife pflegbar gehalten werden. Dadurch würde das Erstellen von diesen erheblich vereinfacht werden. Außerdem müssten Straßen in die Pflege eines Liefergebietes mit aufgenommen werden, da einige Energieversorger diese bis auf Straßenebene untergliedern.

4.1.2. Tarifrechner

Tarifrechner PowerCommerce

In der derzeitigen Version von PowerCommerce ist der Tarifrechner in drei Schritte untergliedert. Zunächst wird der Nutzer aufgefordert die gewünschte Sparte und seinen Kundentyp zu wählen. Als Kundentyp stehen ihm dabei Privat- oder Geschäftskunde zur Auswahl. Für die Bestimmung der Abnahmestelle ist die Angabe der Postleitzahl oder des Ortes notwendig. In Einzelfällen müssen beide Angaben ausgefüllt werden. Dies ist notwendig, wenn eine Postleitzahl für mehrere Orte (Ortsteile) gelten kann oder ein Ort mehrere Postleitzahlen umfasst. Eine Liefergebietsprüfung auf Straßenebene erfolgte bisher noch nicht. Das ist von erheblichem Nachteil, da einige Produkte ihr Liefergebiet bis auf Straßen- oder sogar Hausnummernebene definieren.

Information			
Bei den von Ihnen gemachten Angaben sind folgende Kosten zu erwarten. Falls Sie Bestandskunde sind und zu einem anderen Produkt wechseln möchten, dann nutzen Sie bitte den Prozess Produktwechsel ! Für diesen Prozess ist ein Login erforderlich.			
Gas			
Gasverbrauch	2500 Jahr		
Liefergebiet	01099 Dresden		
Tarif / Gesamtkosten			
	Jahr	Monat	
<input checked="" type="checkbox"/> Kleinverbrauchstarif	215,71 €	17,98 €	<input checked="" type="checkbox"/> Vertrag abschließen <input checked="" type="checkbox"/> Produktwechsel
<input checked="" type="checkbox"/> Grundpreistarif	219,34 €	18,28 €	<input checked="" type="checkbox"/> Vertrag abschließen <input checked="" type="checkbox"/> Produktwechsel
<input checked="" type="checkbox"/> Tarif für Einzelofenheizung	240,17 €	20,01 €	<input checked="" type="checkbox"/> Vertrag abschließen <input checked="" type="checkbox"/> Produktwechsel
Alle aufgeführten Preise sind Brutto-Preise. Alle Angaben ohne Gewähr.			
<input checked="" type="checkbox"/> Zurück			

Abbildung 4.2.: Ergebnisliste Produktrechner PowerCommerce

Wenn für die angegebene Verbrauchsstelle Tarife angeboten werden, wird der Tarifrechner fortgesetzt. Im nachfolgenden Schritt wird der Nutzer gebeten seine Verbrauchsangaben einzugeben. Die Anzeige der Felder sowie die Bezeichnung dieser erfolgt dynamisch aus den zur Verfügung stehenden Tarifen. Dazu werden aus allen Tarifen die Tarifbestandteile herausgefiltert, die eine Nutzereingabe (`userValueFlag = true`, vgl. 4.1.1 Beschreibung der Tariftabellen) darstellen.

4. Konzeption

Im letzten Schritt des Tarifrechners von PowerCommerce wird die Ergebnisliste dargestellt. Im oberen Bereich der Ansicht erhält der Nutzer eine Zusammenfassung seiner eingegebenen Daten zum Verbrauch und dem Liefergebiet. Anschließend werden alle verfügbaren Produkte mit ihrem Namen, dem Jahresgesamtpreis und einem Monatspreis präsentiert. Jeder Eintrag der Liste erhält in einer weiteren Spalte Links zu weiterführenden Prozessen, wie Vertragsabschluss oder Produktwechsel. Die Namen der einzelnen Tarife sind als Link an eine detaillierte Produktbeschreibung geknüpft. Hier erhält der Kunde weitere Angaben zu Laufzeiten oder Kündigungsfristen.

Positiv zu bemerken ist die schlichte und klar strukturierte Erscheinung des Produktrechners. Dadurch wird der Kunde transparent und wertungsfrei an die gewünschten Preisinformationen herangeführt, die er haben möchte.

Die Einfachheit des Produktrechners kann aber auch negativ erachtet werden, da dadurch keine marketingstrategischen Anreize definiert werden können, um den Kunden bei der Suche nach seinem passenden Produkt zu lenken.

Weiterhin ist es unbefriedigend, dass eine Preisbestimmung nur für eine Sparte erfolgen kann. Hier wäre eine Mehrfachauswahl wünschenswert. Dies hätte zur Folge, dass die Ergebnisseite für die Mehrspartenfähigkeit angepasst werden muss. Folglich müsste auch ein anschließender Vertragsabschluss für mehrere Sparten gleichzeitig möglich sein.

Für die Ergebnisseite wäre es von Vorteil, wenn Informationen wie Arbeitspreis, Grundpreis, Laufzeiten oder Beschreibungen dem Kunden nicht erst per Klick auf Details in einem extra Prozess angezeigt werden. Hier wäre es anzustreben, die Informationen beim Klick auf den Tarifnamen direkt unterhalb der bisherigen Zeile des Tarifes angezeigt werden. Damit hätte der Kunde alle Informationen auf einen Blick und kann diese auch auf Wunsch mit denen der anderen Tarife vergleichen.

Ein weiterer Mangel ist die umständliche Neuberechnung, wenn der Kunde seinen Verbrauch noch einmal ändern möchte. Dazu muss er bisher die aktuelle Seite verlassen, um auf die vorherige zurückzukehren. Dort kann er die Eingabe korrigieren und wieder auf die Ergebnisseite wechseln. Für eine Umgestaltung dieses Prozesses wäre eine direkte Änderung der Eingabe auf der Ergebnisseite von Vorteil. Dazu wäre eine Realisierung mit Hilfe von Ajax denkbar, damit nur die Ergebnisliste und nicht die komplette Seite neu geladen werden müssten.

Tarifrechner unabhängiger Anbieter am Beispiel von Verivox und TopTarif

Neben den Tarifrechnern der Energieversorger gibt es auch einige unabhängige Verbraucherportale und Produktvergleichssysteme im Internet. Der bekannteste Anbieter für die Strompreisrecherche ist dabei Verivox. Alternativen dazu sind TopTarif, Preisvergleich⁴,

⁴<http://www.preisvergleich.de/>

u.v.m.

Die meisten dieser Verbraucherportale bieten neben der Berechnung von Stromtarifen noch andere Produkte, wie Gas oder DSL, an.

Exemplarisch für die Tarifrechner unabhängiger Anbieter sollen hier Verivox[VERI] und TopTarif[ToTa] näher betrachtet werden.

Als Einstieg in den Tarifrechner wird der Nutzer gebeten seine Postleitzahl und den Vorjahresverbrauch anzugeben.

Die Eingabe der Postleitzahl ist notwendig, um eine Liefergebietsbestimmung durchzuführen und den für das Gebiet geltenden Grundversorger zu bestimmen. Weiterhin werden aus dieser Angabe der Netzbetreiber und damit die geltenden Netzentgelte bestimmt, welche anschließend in den Tarif eingerechnet werden. Daraus ergibt sich ein großer Nachteil für einige Tarife. Die Liefergebiete einiger Produkte sind nicht nur auf Postleitzahlen oder Orte beschränkt. Hier kann es zu einer Granularität bis auf Straßen- oder Hausnummernebene kommen! Es ist jedoch verständlich, dass eine Tarifrecherche über eine Vielzahl von Versorgern bei zu hoher Detailstufe sehr viel Zeit in Anspruch nehmen würde und eine zügige Informationsbereitstellung für den Nutzer nicht möglich wäre.

Für die Angabe des Vorjahresverbrauchs gibt es in der Regel zwei Varianten. Auf der einen Seite kann man den konkreten Zahlenwert eingeben, welcher auf der Vorjahresrechnung zu finden ist. Auf der anderen Seite kann eine Angabe zur Haushaltsgröße (Personenanzahl) gemacht werden. Anhand üblicher Verbrauchszahlen kann damit ein Rückschluss auf den möglichen Vorjahresverbrauch gezogen werden.

Weiterhin wird der Nutzer dazu aufgefordert eine Auskunft zum Kundentyp zu geben. Zur Auswahl stehen dabei "Privatkunde" oder "Gewerbekunde".

Bei Verivox wird dem Kunden noch die Möglichkeit gegeben eine Angabe zum Nutzungsverhalten während der Nebenzeit⁵ zu machen. Diese ist aber nur für Kunden mit einem Zweitarifzähler interessant.

Nach den persönlichen Angaben, können noch Filtereinstellungen gesetzt werden, um die Ergebnisliste der verfügbaren Tarife noch genauer zu definieren. Dafür stehen Filter, wie Anzahl der Tarife pro Anbieter, nur Ökostrom, Tarife mit Kautions berücksichtigen, u.a., zur Verfügung. Damit kann der Nutzer für sich uninteressante Angebote ausblenden lassen.

Die Ergebnisliste selbst ist in einer Tabelle mit fünf Spalten organisiert. Es werden Informationen zu den Tarifkosten, dem Ersparnis gegenüber eines anderen Tarifs (meist

⁵Für Kunden mit Zweitarifzählern unterscheidet man zwischen einem Hochtarif (6 Uhr bis 22 Uhr) und Niedertarif (22 Uhr bis 6 Uhr).

4. Konzeption

Strom-Preisvergleich

Mit unserem Stromrechner können Sie sich Ihren persönlichen Stromtarif berechnen lassen. Das funktioniert einfach und unkompliziert: Daten eingeben und dann auf "Tarif berechnen" klicken.

Abbildung 4.3.: Verbrauchseingabe Verivox, Quelle: [VERI]

Grundversorger des Gebietes), Anbieter und Name des Tarifes sowie einige Tarifiedetails wiedergegeben. Weiterhin besteht die Möglichkeit sich entweder direkt anzumelden oder die Unterlagen herunterzuladen.

Die Tarifrechner beider Anbieter liefern cursorsensitive Informationen zu einzelnen Tarifbestandteilen. Bei Verivox wird dabei die Ausschrift der Tarifbedingung mit der zusätzlichen Information verknüpft. Toptarif hingegen liefert vor jeder Tarifbedingung ein Informationszeichen, welches beim Überfahren mit der Maus die Zusatzinformationen preis gibt.

Berechnungskriterien

PLZ:

Ort:

Verbrauch (in kWh):

Nutzung:

Tarife pro Anbieter:

Weitere Kriterien

☐ Sonderabschlag einberechnen

☒ Erstjahresrabatt berücksichtigen

☐ Nur monatliche Zahlungsweise

☐ Nur Ökostrom

☐ Nur Tarife mit Preisgarantie

Aktualisieren

Legende

- Detailinformation
- Nur Ökostrom
- Lokaler Versorger
- Tipp
- Erklärung

Die Ersparnis bezieht sich auf den Tarif » ENSO-Strom-PRIVAT « von » ENSO Energie Sachsen Ost AG « [Tarif ändern](#)

Hinweis: Alle Preise sind Brutto-Preise (inkl. MwSt.)

[Detailvergleich \(max. 3\)](#) [Wie lese ich die Ergebnistabelle?](#) [per E-Mail](#) [Druckansicht](#)

	BEITRAG	ERSPARNIS	ANBIETER / TARIF	ANMERKUNGEN	ZUM ANTRAG
1.	<input type="checkbox"/> 349,58 €	281,74 €	FlexStrom AG 2400er Frühlingsaktion, mit Preisgarantie, mit Treuebonus ★★★★★	Paket, 2400kWh Preisgarantie bis zum 31.08.2009 Zahlungsweise: jährlich 100,00 EUR Treue-Bonus	Weiter →
2.	<input type="checkbox"/> 395,58 €	235,74 €	FlexStrom AG 2400er Partner, mit Preisgarantie, mit Treuebonus ★★★★★	Paket, 2400kWh Zahlungsweise: vierteljährlich Preisgarantie bis zum 31.08.2009 Sonderabschlag: 50,00 EUR 90,00 EUR Treue-Bonus	Weiter →

Abbildung 4.4.: Ergebnisliste TopTarif, Quelle: [ToTa]

Ein großer Vorteil der stromanbieterunabhängigen Tarifrechner liegt in der Bewertung der einzelnen Versorger durch die Nutzer. So haben neue Kunden die Möglichkeit sich vorab ein Bild über die Qualität der Dienstleistungen des EVU zu machen. Toptarif

bietet die Auswertung der Beurteilung direkt für jeden Tarif als eine Fünf-Sterne-Skala an. Bei Verivox ist der Name des Versorgungsunternehmens verlinkt und liefert eine Seite mit vielen Informationen zum diesem. Die Auswertung der Beurteilung erfolgt hier ebenfalls über eine Fünf-Sterne-Skala.

Ein weiteres Plus stellt der direkte Tarifvergleich mehrerer Anbieter dar. Dazu wählt der Nutzer sich die für ihn interessanten Tarife mit Hilfe von Checkboxen aus (bei Top-tarif maximal drei) und führt anschließend einen Tarifvergleich durch. Dabei werden sämtliche Tarifbedingungen, wie Zahlungsart, Laufzeit und Energiemix gegenüber gestellt. Diese Methode fasst dem Kunden die für ihn attraktiven Punkte noch einmal auf einen Blick zusammen.

Nachteilig, aus Sicht des einzelnen EVU, ist die grafisch einfache Darstellung der Tarife. Die Rechner bieten lediglich eine Auflistung von Fakten und stellen diese in einem schlichten Format dar. Einzelne Grafikelemente, wie Symbole für Ökostrom oder die Anzeige des Grundversorgers, sind die einzigen Alternativen zu der textuellen Beschreibung des Tarifes. Marketingstrategische Werkzeuge, wie die Hervorhebung von Schlagworten oder der Einsatz einer Grafik für den Tarif als Blickfang für den Kunden erfolgt hier nicht. Die Tarifrechner schaffen damit eine gleichwertige Darstellung aller Tarife und Anbieter.

Zusammenfassend kann festgehalten werden, dass die Tarifrechner unabhängiger Anbieter ähnlich aufgebaut sind und dem Kunden eine sachliche, wertungsfreie Darstellung der Tarife nach seinen Eingaben und Filtereinstellungen liefert. Positiv ist die Option der Bewertung, sowie der Tarifvergleich hervorzuheben. Für die einzelnen Versorgungsunternehmen ist, aus Marketingsicht, die schlichte Darstellung der Produkte negativ festzuhalten.

Tarifrechner von Energieanbietern am Beispiel von ENSO und Energie SaarLorLux

Einige Energieversorger bieten auf ihren Internetseiten einen Tarifrechner für ihre Kunden an. Anders als bei unabhängigen Anbietern, berücksichtigen diese ausschließlich die eigenen Tarife.

Bereits beim Einstieg in den Rechner können die Energieversorger auf die vorherrschende Geschäftslogik eingehen.

So bietet ENSO, neben der Angabe der Postleitzahl und des Verbrauches, den Kunden die Möglichkeit sich zwischen Ein- und Zweitarifzählern zu entscheiden. Je nach dem eingebauten Zähler können anschließend die entsprechenden Tarife dargestellt werden. Die Verbrauchsangabe muss für ENSO als konkreter Zahlenwert erfolgen. Hilfreich ist dabei das cursorsensitive Informationsfeld, welches beim Darüberfahren eine Liste möglicher Verbrauchswerte anhand der Haushaltsgröße angibt. Die Ergebnisliste ist bei ENSO

4. Konzeption

schlicht organisiert. Im oberen Teil sind die getätigten Eingaben noch einmal zu sehen und können jederzeit angepasst werden.

Produktvergleich
Finden Sie Ihr passendes ENSO-Produkt

Postleitzahl:
Zählertyp:
Jahresverbrauch: kWh

Service-Telefon
0180 1 686868
3,9 ct pro angefangene Minute aus dem Festnetz der DT AG, abweichende Preise aus dem Mobilfunknetz möglich

Stromkennzeichnung
☒ Der Strom-Mix der ENSO Energie Sachsen Ost AG

ENSO-Strom-DIREKT
597,68 EUR/Jahr
☒ Ich bin für den Direkt-Service registriert.
☐ Ich bin noch nicht für den Direkt-Service registriert.
☐ Ich bin noch kein ENSO-Kunde.

ENSO-Strom-SWING
609,58 EUR/Jahr
☒ Ich bin für den Direkt-Service registriert.
☐ Ich bin noch nicht für den Direkt-Service registriert.
☐ Ich bin noch kein ENSO-Kunde.

ENSO-Strom-UMWELT
629,83 EUR/Jahr
☒ Ich bin für den Direkt-Service registriert.
☐ Ich bin noch nicht für den Direkt-Service registriert.
☐ Ich bin noch kein ENSO-Kunde.

Verfeinern Sie Ihr Ergebnis
Filtern nach Produkten
► mit Ökostrom
► mit Preisgarantie
► ohne Einzusermächtigung
► nur online
Filtern nach Kündigungsfristen
► 1 Monat
Filtern nach Mindestlaufzeit
► 6 Monate
► keine

Informationen zu ENSO-Strom-UMWELT
Umweltfreundlicher Strom für Haushaltsbedarf bei Anschluss im Niederspannungsnetz im Stromnetzgebiet der ENSO Netz GmbH.
Zähler:
Für direkt messende Drehstromzähler (Eintarifzähler).
Preise ab 01.01.2009:

	Einheit	netto	brutto
Verbrauchspreis	ct/kWh	18,07	21,50
Grundpreis	€/Jahr	77,59	92,33

Der Verbrauchspreis enthält die gesetzliche Stromsteuer in voller Höhe, zzt. 2,05 ct/kWh netto. Das Stromentgelt wird auf Basis von Nettopreisen ermittelt und erhöht sich um die jeweils gültige Umsatzsteuer (zzt. 19 %).

Abbildung 4.5.: Ergebnisliste ENSO, Quelle: [ENSO]

Anschließend erfolgt eine Auflistung der vorhandenen Tarife. Dabei werden alle Produkte auf die gleiche Art und Weise präsentiert. Dem Kunden wird der Name des Tarifes sowie der Preis für den angegebenen Verbrauch dargestellt. Weitere Tarifdetails sind auf den ersten Blick nicht ersichtlich. Für diese Informationen ist eine cursorsensitive Schaltfläche notwendig. Beim Überfahren der Schaltfläche “Mehr Produktinfos” mit der Maus, wird ein neuer, farbig hervorgehobener Bereich sichtbar. Darauf sind eine knappe Beschreibung, die zugelassenen Zählerarten und die gültigen Preise erkenntlich.

Neben dem Namen und dem Preis enthält jeder Tarifeintrag der Auflistung noch eine Auswahlliste von drei Elementen. Damit kann der Kunde, sofern er das Produkt bestellen möchte, eine Auswahl zu seinem derzeitigen Kundenstatus gegenüber der ENSO treffen. Es stehen die Alternativen “Direktservicekunde (bereits Nutzer des Onlineportal)”, “kein Direktservicekunde (Kunde der ENSO ohne Nutzung des Onlineportal)” und “Neukunde” zur Verfügung. Gestartet wird die Bestellung über den sich darunter befindlichen Schalter.

Eine grafische Hervorhebung einzelner Tarife oder die Kenntlichmachung spezieller Produktvorteile werden bei der ENSO nicht genutzt.

4. Konzeption

Rechts neben der Auflistung der vorhandenen Produkte kann der Kunde verschiedene Filtereinstellungen aktivieren. Damit ist eine feinere Eingrenzung der Produktpalette möglich.

Auch bei der Energie SaarLorLux AG ist der Einstieg in den Tarifrechner an die vorherrschende Geschäftslogik angepasst. Der Kunde wird zunächst aufgefordert drei Angaben zu übermitteln. Die erste ist die Postleitzahl, welche auch hier zur Liefergebietsprüfung herangezogen wird. Als Zweites wird geprüft, ob der Nutzer bereits Kunde der Energie SaarLorLux AG ist. Diese Information wird lediglich zur Aus- und Einblendung eines Bonizahlungsfeldes genutzt. Der letzte Eingabewert ist der durchschnittliche Jahresverbrauch des Kunden. Dieser kann als konkreter Zahlenwert oder als Angabe der Haushaltsgröße erfolgen.

Ihr günstigster Tarif bei 2500 kWh/Jahr

Preise gültig ab: 01.01.2009 (Ausnahme BestPlus gültig ab: 01.02.2009)

 591,99 €/Jahr* Unsere Tarifempfehlung	Grundpreis/Jahr netto: 89,91 € brutto: 106,99 € Verbrauchspreis/kWh netto: 17,65 ct brutto: 21,00 ct	<ul style="list-style-type: none">100% Ökostrom ohne AufpreisLaufzeit: 1 MonatVerlängerung: 1 MonatKündigungsfrist: 1 Monat	<input type="checkbox"/> Jährliche Zahlungsweise <input type="checkbox"/> RegioFonds <input checked="" type="checkbox"/> 40 € Prämie Vertrag herunterladen Online anfordern
---	---	--	---

Weitere Tarife

	600,51 €/Jahr*	Für weitere Infos klicken Sie bitte auf den Pfeil 
---	-----------------------	---

Abbildung 4.6.: Ergebnisliste Energie SaarLorLux, Quelle: [SaLoLu]

In der Ergebnisliste des Tarifrechners ist die Unterscheidung der einzelnen Tarife mit Hilfe einer Grafik hervorzuheben. Weiter fällt auf, dass der günstigste Tarif nicht nur als Erster erscheint, sondern auch durch einen Text auf den Preisvorteil hingewiesen wird. Das preiswerteste Produkt wird sofort mit allen Tarifdetails sichtbar dargestellt. Alle anderen Tarife werden nur mit der entsprechenden Grafik und dem ermittelten Preis gezeigt. Um weitere Details zu diesen Tarifen zu sehen, ist ein Klick auf einen Schalter notwendig. Dabei ist zu bemerken, dass die Betätigung des Schalters ein JavaScript-Event auslöst, um die Informationen einzublenden. Einem Nutzer ohne JavaScript-Unterstützung im Browser wird keine Alternative zu diesem Schalter angeboten. Ihm bleiben die Tarifdetails der anderen Tarife verborgen.

In der Auflistung der Tarife wird jedes einzelne Produkt nach dem gleichen Schema dargestellt. Der dabei zur Verfügung stehende Anzeigebereich wird in vier Spalten aufgeteilt.

In der ersten Spalte wird das Produkt mit der Grafik und dem ermittelten Gesamtpreis präsentiert.

4. Konzeption

Rechts davon werden Preisdetails aufgezeigt. So erfährt der Nutzer, welchen Grundpreis er für das Jahr bezahlen würde und mit welchem Verbrauchspreis er pro Kilowattstunde zu rechnen hat.

Die darauffolgende Spalte listet stichpunktartig geltende Tarifiedetails auf. Diese können, je nach Tarif, in Anzahl und priorisierter Reihenfolge erfolgen.

Im letzten Bereich der Anzeige werden preisbeeinflussende Tarifoptionen mit Hilfe von Checkboxen zur Auswahl gestellt. Beispielsweise kann ein Neukunde sich eine Einmalzahlung für das erste Jahr gutschreiben lassen. Informationen zu den Tarifoptionen liefert eine cursorsensitive Hinweisgrafik hinter jedem Eintrag. Tarifoptionen die einen direkten Einfluss auf den Gesamtbetrag oder den Verbrauchspreis haben, werden in Echtzeit angepasst. Damit kann die Veränderung des Preises direkt vom Nutzer wahrgenommen werden. Nachteilig ist, sowohl für die Hinweise zu den Tarifoptionen als auch die direkte Preisanpassung in Echtzeit, eine fehlende Alternative für Nutzer ohne Javascript.

Die aufgeführten Betrachtungen betrafen ausschließlich die Tarifrechner der ENSO Energie Sachsen Ost AG und Energie SaarLorLux AG. Aber auch für die vielen anderen Energieversorger ist festzustellen, dass die Nutzung eines eigenen Tarifrechners an die jeweils vorherrschende Geschäftslogik angepasst werden kann. Eine marketingstrategische Hervorhebung einzelner Tarife ist möglich. Damit können Kunden gezielt auf ein bestimmtes Produkt aufmerksam gemacht werden.

Als entscheidender Vorteil der beiden Tarifrechner gegenüber der Ausprägung von PowerCommerce liegt in der schnellen Anzeige zusätzlicher Informationen zu einem bestimmten Produkt. Bei ENSO hält der Kunde lediglich den Mauszeiger über eine Schaltfläche *Produktinfos* und eine detaillierte Beschreibung des Produktes wird dem Kunden ersichtlich. Bei SaarLorLux wird durch einen Klick auf den entsprechenden Tarif ein neuer Bereich aufgeklappt, um dem Kunden weiterführende Angaben zu liefern.

Da in beiden Fällen der Produktrechner mit JavaScript arbeitet und meist keine alternativen Wege dessen Funktionalität ersetzen, ist dies als enorme Schwachstelle zu erkennen. Hier sollte wenigstens der Hinweis erscheinen, dass ein Kunde die Skriptsprache in seinem Browser aktivieren sollte, um den vollen Funktionsumfang nutzen zu können.

4.1.3. Produktkonfigurator

Im Administrationsbereich von PowerCommerce hat der Mitarbeiter des EVU die Möglichkeit, Tarife zu erstellen und anschließend zu bearbeiten. Der Produktkonfigurator gliedert sich dabei in drei Teilprozesse.

Im Prozess „Produkt anlegen“ kann der Bearbeiter einen Tarif mit seinen allgemeinen Informationen erstellen. Hier wird die zu bedienende Sparte angegeben und die Kundengruppe, für die der Tarif zur Verfügung stehen soll, ausgewählt. Weiterhin wird darüber entschieden, ob der Tarif nach dem Erstellen aktiviert oder deaktiviert sein

4. Konzeption

soll. Deaktivierte Tarife werden nicht im Tarifrechner dargestellt und stehen nicht beim Vertragsabschluss zur Auswahl. Weitere allgemeine Angaben sind der Name des Tarifes sowie die Angabe zum Tariftyp. Ob ein Produkt die persönliche Unterschrift des Endkunden für einzelne Prozesse benötigt, wird in den nachfolgenden Eingabefeldern abgefragt. Abschließend können Tarifattribute und sprachabhängige Daten für den Tarif hinterlegt werden.

Produktverwaltung - Produkt anlegen Schritt 1

Information

Alle mit * gekennzeichneten Felder müssen angegeben werden.

Allgemeine Angaben

Sparte:

Tarif für:

Aktiviert: ☒ Ja ☐ Nein

Weitere Angaben

Tarifkürzel (CS.VA): *

Tariftyp: * ☒ Eintariffzähler ☐ Zweitarriffzähler

Unterschrift bei Vertragsabschluss: * ☐ Ja ☒ Nein

Unterschrift bei Ummeldung: ☐ Ja ☒ Nein

Unterschrift bei Produktwechsel: ☐ Ja ☒ Nein

Vertragsabschluss über statisches PDF: ☐ Ja ☒ Nein

Tarifattribute

Fügen Sie hier Tarifattribute hinzu. Bearbeiten

Sprachabhängige Daten hinterlegen

Geben Sie hier die Produktbezeichnung und die Dokumenten-URLs der Vertragsgrundlagen (Sparte und Produkt) an. Zu einem aktivierten Produkt muss mindestens ein Dokument verlinkt sein. Deutsch

Abbildung 4.7.: Prozess „Produkt anlegen“ im PowerCommerce

Der Prozess „Produkt ändern“ bietet die Möglichkeit die Angaben aus dem Bearbeitungsschritt „Produkt anlegen“ zu editieren. Um einen Tarif zu bearbeiten, muss er zunächst ausgewählt werden (Abbildung 4.8 Seite 35). Dazu wird in zwei vorgeschalteten Prozessschritten die Sparte und die Kundengruppe des Tarifes abgefragt. Weiterhin muss der Nutzer angeben, ob der zu wählende Tarif aktiviert oder deaktiviert ist. Danach werden alle auf die Angaben passenden Tarife in einer Liste ausgegeben. Durch Anwählen des zu ändernden Tarifes gelangt man in den eigentlichen Bearbeitungsprozess. Dieser präsentiert sich optisch identisch zum Prozess „Produkt anlegen“ (vgl. Abbildung 4.7 Seite 34).

Die umfangreichste Routine zur Produktkonfiguration ist die Möglichkeit einen Tarif zu erweitern. Dabei werden, neben den allgemeinen Informationen, Tarifbestandteile, die

4. Konzeption

The screenshot shows a web form titled 'Allgemeine Auswahl'. It contains the following elements:

- Sparte:** A dropdown menu with 'Strom (S)' selected.
- Tarif für:** A dropdown menu with 'Privat- und Geschäftskunden' selected.
- Aktiviert:** Two radio buttons, 'Ja' (selected) and 'Nein'.
- Produktauswahl:** A section with a checkbox for 'Ökostrom (ÖKOSTROM)'.
- Zurück:** A button at the bottom left.

Abbildung 4.8.: Produktauswahl im Prozess „Produkt ändern“ im PowerCommerce

Tarifformel und das Liefergebiet zugeordnet. Sie gliedert sich in insgesamt acht Bearbeitungsschritte. Auf den Seiten eins und zwei erfolgt die Auswahl des zu erweiternden Tarifes. Die Umsetzung der Auswahl ist dabei analog zum Änderungsprozess implementiert.

Im eigentlichen Erweiterungsprozess kann der Bearbeiter zunächst Angaben zum Kundentyp und der Aktivität des Tarifes machen. Weiterhin kann ein tarifspezifisches Bild eingefügt werden. Ob für diverse Prozesse eine Unterschrift des Endkunden benötigt wird, kann auch auf dieser Seite eingestellt werden. Es folgt eine Auswahl der zulässigen Zahlungsmodalitäten, die für diesen Tarif gelten sollen. Abschließend werden Laufzeiten und Kündigungsfristen des Tarifes abgefragt.

Im darauffolgenden Schritt können Tarifbestandteile angelegt und bestehende geändert werden. Die Tarifbestandteile werden später in die Tarifformel zur Preisbestimmung eingebaut. Tarifteile können fixe Preise (Grundpreis, Arbeitspreis) oder Nutzereingaben (Verbrauch) sein.

In Schritt 3 können die zuvor angegebenen Tarifbestandteile in eine Tarifformel eingebaut werden (Abbildung 4.9 Seite 36). Dazu werden die erstellten Tarifteile im oberen Bereich der Anzeige noch einmal aufgelistet. Da bestehende Tarifformeln wiederverwendet werden können, wird eine Liste aller im System verfügbaren Tarifformeln angegeben. Wenn eine bestehende Tarifformel auf die für den Tarif definierten Bestandteile passt (alle Tarifteile kommen in der Formel vor), kann diese für den Tarif übernommen werden. Eine bestehende, nicht passende wird grau dargestellt und kann nicht übernommen werden. Wenn eine neue Tarifformel angegeben werden soll, muss ein eindeutiger Formelname, sowie die eigentliche Formel angegeben werden. Es ist dabei darauf zu achten, dass in der Tarifformel die Namen der erstellten Tarifbestandteile verwendet werden.

Das Liefergebiet wird in dem nächsten Bearbeitungsschritt definiert. Es besteht aus einer Kombination von ein- und ausgeschlossenen Bundesländern, Landkreisen, Postleitzahlen und Orten. Damit können beispielsweise aus einem Bundesland einzelne Postleitzahlen ausgeschlossen werden.

4. Konzeption

The screenshot displays the 'Eingegebene Tarifparts' (Entered Tariff Parts) section of the PowerCommerce interface. It features a table with two columns: 'Eingegebene Tarifparts' and 'Vordefinierte Attribute'. The table lists 'AchievementPrice', 'ConsumptionPrice', and 'PowerConsumption (UserValue)' with a value of '- / -'. Below this is the 'Verfügbare Tarifformeln' (Available Tariff Formulas) section, which lists 'Power_OneMeter' and 'Power_TwoMeter'. 'Power_OneMeter' has buttons for 'Anzeigen' (Show) and 'Übernehmen' (Take Over), while 'Power_TwoMeter' has an 'Anzeigen' button. The 'Details' section at the bottom shows the 'Tarifformelname' (Tariff Formula Name) as 'Power_OneMeter' and the 'Tarifformel' (Tariff Formula) as 'result = AchievementPrice + (PowerConsumption * ConsumptionPrice);'. Navigation buttons 'Zurück' (Back) and 'Weiter' (Next) are at the bottom.

Eingegebene Tarifparts	Vordefinierte Attribute
AchievementPrice	
ConsumptionPrice	
PowerConsumption (UserValue)	- / -

Verfügbare Tarifformeln	
Power_OneMeter	Anzeigen Übernehmen
Power_TwoMeter	Anzeigen

Details

Hilfe ?

Tarifformelname:

* Power_OneMeter

Tarifformel:

* result = AchievementPrice + (PowerConsumption * ConsumptionPrice);

Zurück

Weiter

Abbildung 4.9.: Tarifformel im Prozess „Produkt erweitern“ im PowerCommerce

Im vorletzten Prozessschritt werden Übersetzungen zum Tarif und dessen Bestandteilen angegeben.

Den letzten Schritt stellt eine Zusammenfassung der angegebenen Daten dar. Hier hat der Bearbeiter noch einmal die Möglichkeit seine Angaben zu prüfen.

Nach einer Analyse des derzeitigen Produktkonfigurators sind einige Punkte als verbesserungswürdig aufgefallen.

Insgesamt betrachtet, ist die Prozedur, um einen Tarif zu erstellen, viel zu lang und zu komplex. Hier wäre es wünschenswert, dass der Kunde auch zwischendurch den Prozess abschließen kann. Denn möchte er nur einen Preis des Tarifes anpassen, so muss er dennoch den gesamten Ablauf durchlaufen, um die Routine erfolgreich beenden zu können. Eine Teilung des Prozesses in kleinere Einzelprozesse wäre hier ratsam.

Weiterhin sollte die Pflege eines Liefergebietes an zentraler Stelle für mehrere Tarife erfolgen. Denn in der Regel wird eine Vielzahl der angebotenen Tarife das selbe Gebiet bedienen und dann ist es unnötig dieses für jeden Tarif neu zu erstellen.

Es sollte außerdem die Möglichkeit geben die Tarifbestandteile separat und global für alle Tarife zu pflegen. Denn bisher muss ein Tarifbestandteil *Stromverbrauch* für jeden Stromtarif neu angelegt werden. Diese Informationen sind redundant und unnötig.

4.1.4. Schnittstellen

In der aktuellen Implementierung von PowerCommerce wird der Tarifimport mit Hilfe einer XML-Datei realisiert. Diese enthält einzelne Tarifelemente in der Objektstruktur des Tarifmodells. Beim Einlesen der Daten wird für jedes Tarifelement ein neues Datenbankobjekt erzeugt. Bereits verfügbare Angaben, wie Sparte oder Postleitzahl, werden nicht neu angelegt. Hier erfolgt ausschließlich eine Referenzierung der Objekte mit Hilfe eines Fremdschlüssels.

Technisch realisiert wird das Einlesen der Importdatei mit Hilfe eines DOM-Parsers. Dabei wird das gesamte Dokument zunächst eingelesen, in eine Baumstruktur umgewandelt, um sich anschließend daran entlang zu hangeln. Dabei werden die einzelnen Objektknoten eingelesen und in die Datenstruktur geschrieben.

Die Nutzung des Tarifmodells ist nur über die Weboberfläche von PowerCommerce implementiert. Eine Nutzung durch weitere Anwendungen, wie es durch WebServices möglich wäre, ist nicht realisiert.

Um Tarife aus PowerCommerce zu entnehmen, besteht die Möglichkeit eines Tarifexports. Dabei wird aus den, in der Datenbank verfügbaren Daten eine XML-Datei erzeugt. Diese kann bei Bedarf wieder in das System importiert werden.

Nachteilig an diesem System ist eine fehlende Schnittstelle, um die Berechnungsmethoden des Produktrechners auch außerhalb des PowerCommerce nutzen zu können. Es wäre vorteilhaft, wenn der Kunde die Preisberechnung von PowerCommerce auf seiner eigenen Webseite nutzen könnte. dazu sendet er die eingegebenen Daten beispielsweise an einen Webservice im PowerCommerce, dieser durchläuft die Produktrechnerroutine und liefert anschließend das Ergebnis an die Webseite des Kunden zurück.

Ein weiterer Nachteil der derzeitigen Implementierung ist die Verwendung eines DOM-Parsers zum Einlesen der Importdateien. Da es oft zu einer Vielzahl von Tarifen kommt, wäre eine ereignisbasierte Abarbeitung der Dateien wesentlich effektiver und schneller. Denn dabei muss nicht das gesamte Dokument eingelesen werden, sondern es wird sofort Element für Element durchlaufen.

4.2. Soll-Kriterien

Im Laufe der Zeit haben sich einige Schwachstellen des derzeitigen Tarifmodells herausgestellt. Weiterhin macht die Kreativität bei der Tarifgestaltung der Kunden und die Einführung von SmartMetering eine Umgestaltung des Tarifmodells unvermeidlich. Damit geht eine Neuentwicklung des derzeitigen Tarifrechners einher.

Die Anforderungen gliedern sich dabei in vier verschiedene Teilbereiche:

- Allgemein
- das Datenmodell (umfasst die Änderungen auf Datenbankebene)
- der Tarifrechner (aus Sicht des Endkunden - Verbrauchers)
- und ein Produktkonfigurator (aus Sicht des Energieversorgers).

4.2.1. Allgemein

Der Einsatz der entstehenden Applikation soll unabhängig von PowerCommerce einsetzbar sein. Dabei sind mögliche Schnittstellen zu Drittsoftware (z.B. Firmenwebseite) zu berücksichtigen. Eine Integration in das Anwendungspaket von PowerCommerce wird als unproblematisch vorausgesetzt.

4.2.2. Datenmodell

Eine wesentliche Anforderung an das Datenmodell ist ein hohes Maß an Flexibilität. Es soll möglich sein, eine Vielzahl von Tarifszenarien abzubilden, ohne nachfolgende Anpassungen an der Datenstruktur vornehmen zu müssen.

In diesem Zusammenhang wird erwartet, dass Tarifbestandteile nicht mehr direkt mit einem Tarif verknüpft werden. Ein einmal angelegter Bestandteil soll problemlos wiederverwendbar sein. Weiterhin sollen globale Tarifteile, wie Steuern, definiert werden können.

Bestehende Tarife und deren Bestandteile sollen kopierbar sein. Damit muss ein Element nicht in allen Punkten neu erstellt werden, wenn sich nur ein Wert ändert.

Es wird erwartet, dass Tarife und ausgewählte Bestandteile in einer Historie verwaltet werden können. So kann zu jeder Zeit die Entwicklung eines solchen analysiert werden. Ein Tarif kann verschiedenste Dokumente benutzen. Anzahl und Art der Dokumente sollen dabei frei wählbar sein.

Nicht nur die Eingabe des Verbrauchswertes soll preissensitiv sein. Zahlungsart, -intervall und -zeitpunkt sind nur einige Faktoren, die als preisbeeinflussend zu realisieren sind. Eine weitere Anforderung sind Import und Export von Tarifdaten. Dabei ist eine direkte Kommunikation mit dem jeweiligen Abrechnungssystem oder das Nutzen einer zwischengelagerten Datei denkbar.

Das Datenmodell soll spartenübergreifend einsetzbar sein. Das Augenmerk liegt dabei auf den derzeit noch untypischen Sparten für Energieversorger, wie DSL oder ähnliche.

4.2.3. Tarifrechner

Der Kunde soll nach Eingabe seiner Verbrauchsstelle die für ihn günstigsten Tarife angezeigt bekommen. Eine Filterung der Ansicht soll die Auswahl der Tarife nach Kundenwünschen entsprechend optimieren können. Mögliche Filter sind Verbrauchsgrenzwerte oder Tarifattribute, wie die Kennzeichnung als Ökostrom.

Die Anzeige der Tarife soll übersichtlich und klar strukturiert sein. Der Einsatz optischer Besonderheiten bei einzelnen Tarifen ist denkbar.

Für variable Tarifbestandteile soll der Kunde entsprechende Werte auswählen können, um den individuellen Preis ermitteln zu können.

Der Tarifrechner kann als Einstiegspunkt in verschiedene Folgeprozesse dienen. Der Vertragsabschluss sowie der Tarifwechsel sind dabei die wahrscheinlichsten Anwendungsfälle.

4.2.4. Produktkonfigurator

Für den Mitarbeiter des Energieversorgungsunternehmens wird im Backoffice die Möglichkeit bestehen, Tarife schnell und unkompliziert zu verwalten. Die Erstellungsroutine soll dabei gegenüber der jetzigen, soweit möglich, verkürzt werden.

Beim Erstellen von Tarifen und Bestandteilen soll auf die Bestehenden zurückgegriffen werden können, um diese zu kopieren und je nach Bedarf anzupassen.

Beim Einrichten eines neuen Tarifes ist eine modulare Vorgehensweise wünschenswert. So kann die Abfolge der Arbeitsschritte vom Kunden gewählt werden.

Für Tarifoptionen sind Vorgabewerte zu definieren. Globale Tarifbestandteile können unabhängig vom Tarif gepflegt werden.

Eine wesentliche Anforderung stellt die Erstellung einer Wechselmatrix zwischen den einzelnen Tarifen dar. Damit soll festgelegt werden können, von welchem Tarif in einen anderen gewechselt werden kann. Dabei können verschiedene Bedingungen, wie eine geänderte Kündigungsfrist, gelten.

4.3. Konzeption

4.3.1. Datenmodell

Ein Hauptanliegen der Überarbeitung des Tarifmodells ist die problemlose Wartung und Neuentwicklung einzelner Tarifelemente. Weiterhin sollen Tarifbestandteile global pflegbar und mehrfach verwendbar sein. Dies gilt auch für Tarifformeln. Einmal angelegte Formeln sollen in anderen Tarifen verwendbar sein. Wenn an diesen Änderungen vorgenommen werden, soll es die Möglichkeit geben, sofort alle Formeln der entsprechenden

Tarife anzupassen.

Um das neue Datenmodell modularer zu gestalten, wurde viel mit Aggregationstabellen gearbeitet. Dadurch kann eine m:n-Beziehung zwischen den Elementen erstellt werden. Ein weiterer Vorteil liegt darin, einzelne Elemente mehrfach verwenden zu können. Um eine Historie der Tarife, des Liefergebietes, der Tarifbestandteile, sowie der Formeln zu ermöglichen, werden jeweils Spalten für einen Gültigkeitsbeginn (*validFrom*) und ein Gültigkeitsende (*validTo*) sowohl in den Aggregationstabellen, als auch in den Objekttabellen selbst angelegt. Stellt man die Gültigkeit eines Objektes direkt ein, hat dies Auswirkungen auf alle weiteren damit verknüpften Elemente. Soll nur für einen Tarif die Gültigkeit eines verbundenen Elementes geändert werden, werden dafür die Werte der Aggregation angepasst.

Das Liefergebiet wurde völlig überarbeitet und wird nun als eigenständiges Objekt *TariffArea* verwaltet. Die Elemente eines Liefergebietes, wie Postleitzahlen, Orte und Straßen werden als Elemente des Liefergebietes in der Tabelle *TariffAreaEntry* gespeichert. Mit Hilfe der Aggregationstabelle kann ein Liefergebiet nun einmalig erstellt werden und für viele weitere Tarife verwendet werden (siehe dazu Abbildung 4.10 auf Seite 42).

Weiterhin fällt auf, dass einen wesentlichen Anteil die Tarifbestandteile einnehmen. Diese können in Zukunft global gepflegt und per Aggregationstabelle an mehrere Tarife geknüpft werden. Die Unterscheidung, ob ein Tarifpart einen festen Preisbestandteil darstellt oder eine Nutzereingabe repräsentiert, wurde bisher über ein boolisches *user-Value*-Flag gelöst. Im neuen Datenmodell werden die Bestandteilstypen als Integerwert gespeichert. Dadurch ist die Einführung eines weiteren Typs ohne neue Anpassungen möglich.

Eine weitere Neuerung der Tarifbestandteile stellen die Tarifoptionen dar. Diese werden als normaler *TariffPart* angelegt und erhalten zusätzlich Optionswerte. Sie werden in der Tabelle *TariffOptionValue* gespeichert. Damit die einzelnen Werte einer Tarifoption auch einen Einfluss auf den Preis haben können, muss die Tabelle *Price* angepasst werden. Hier wird die Spalte zur Speicherung des Primärschlüssels des zugehörigen Tarifbestandteils umbenannt in *ownerID*, um auch den Optionswerten Preise zuzuordnen zu können. Diese Änderung ist in den Abbildungen nicht aufgeführt.

Die Tarifformeln konnten bisher schon für mehrere Tarife verwendet werden, da an einem solchen nur eine Referenz zum Formelobjekt gehalten wurde. Diese Funktionalität wurde angepasst und auch als Aggregationstabelle realisiert. Damit ist es möglich, einem Tarif auch mehrere Formeln zuzuordnen. Es wäre ja denkbar, dass ein EVU zum Jahreswechsel eine neue Berechnungsgrundlage ihrer Tarife erstellen möchte. Mit dem neuen Modell könnten dafür einem Tarif zwei Formeln zugeordnet werden, die sich in ihrem Gültigkeitszeitraum unterscheiden. So würde die Preisermittlung automatisiert am gewünschten Termin umschalten.

4. Konzeption

Eine weitere Änderung an der Tarifformeltabelle ist das Wegfallen der Bytecodespalte für die Formel. Bisher wurde beim Erstellen einer Tarifformel eine Objektklasse generiert und als Bytecode in der Tabelle abgelegt. Wenn die Formel zur Berechnung herangezogen werden musste, wurde die Klasse instanziiert und die entsprechenden Berechnungsmethoden aufgerufen. Dieses Verfahren war über viele Jahre hinweg im PowerCommerce gewachsen und führte immer wieder zu Unklarheiten, da es nicht möglich war, während der Entwicklung diese Methoden zu debuggen.

Für die neue Datenumsetzung wird eine Evaluierung des Formeltextes mit Hilfe einer Skriptsprache angestrebt. Dabei wird die Programmier- und Skriptsprache *Groovy* zum Einsatz kommen, da diese schon an anderen Stellen im neuen PowerCommerce Verwendung findet. Der Formeltext wird als Skript interpretiert und erhält als Übergabeparameter die zu ersetzenden Variablenwerte. Als Rückgabe erhält man das berechnete Ergebnis.

4. Konzeption

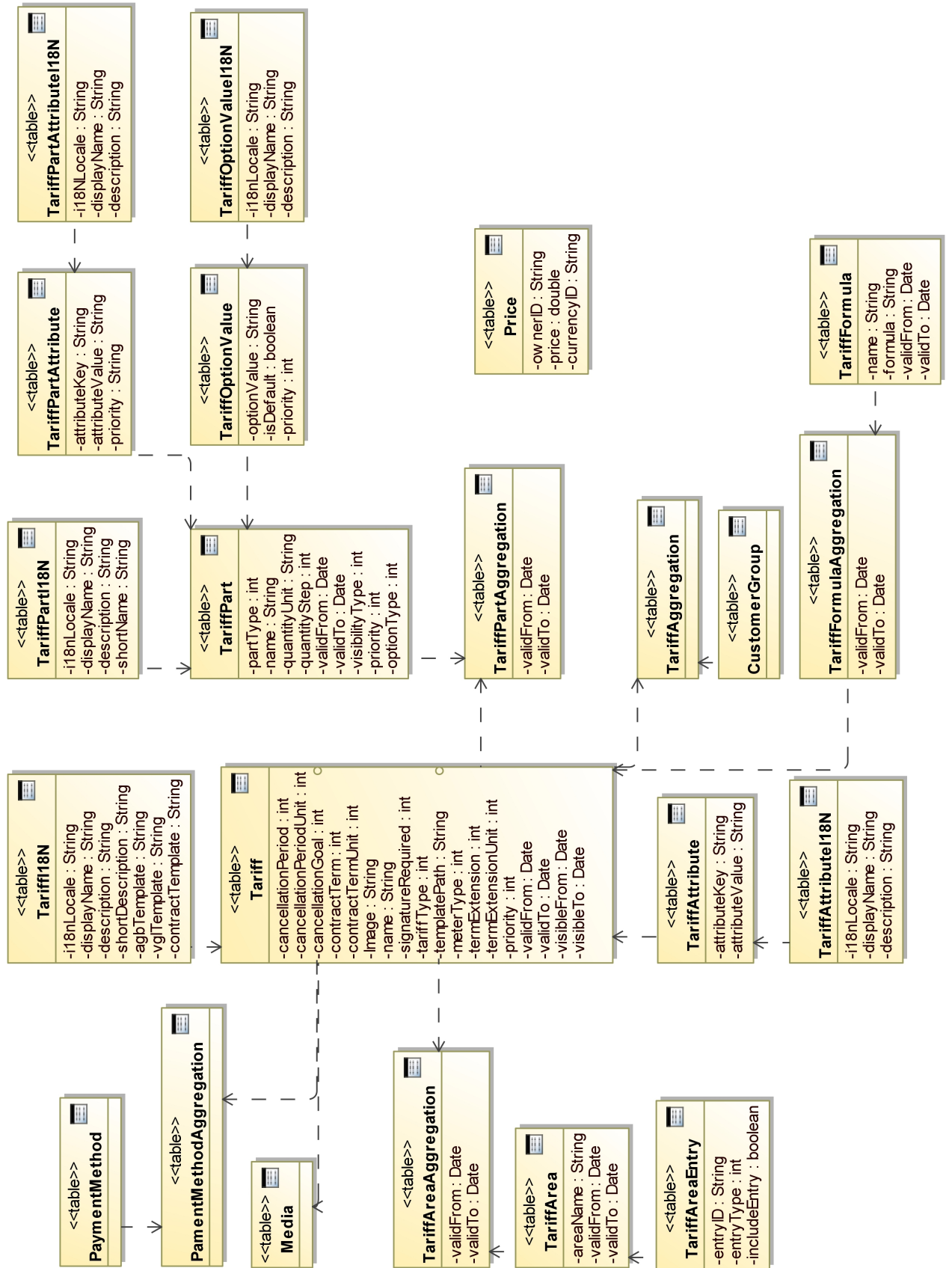


Abbildung 4.10.: Neues Tarifmodell

4.3.2. Tarifrechner

Im Tarifrechner wird der Nutzer aufgefordert, seine Abnahmestelle einzugeben, die gewünschte Sparte und den Kundentyp zu wählen.

Nach dem Abschließen der Eingabe, wird im Prozess geprüft, ob die angegebene Abnahmestelle existiert. Das heißt, es wird recherchiert, ob die Postleitzahl existiert und der Ort für die Postleitzahl verfügbar ist. Falls die Daten vorhanden sind, kann noch eine Zuordnung der Straße zum Ort validiert werden. Wird bei der Prüfung der Abnahmestelle festgestellt, dass sie nicht bekannt ist, bekommt der Kunde eine Fehlerausschrift und wird aufgefordert seine Angaben zu berichtigen.

Existiert die Verbrauchsadresse wird im zweiten Schritt ein Abgleich mit den Liefergebieten der Tarife gemacht. Wenn für die Adresse keine Tarife angeboten werden, kann der Tarifrechner beendet werden oder der Kunde wird benachrichtigt, dass seine Lieferanschrift nicht im Versorgungsgebiet liegt. Daraufhin wäre es denkbar einen alternativen, überregionalen Versorger zu empfehlen. Anderenfalls hat der Nutzer die Möglichkeit seine Daten zu ändern. Wie der Prozess im Einzelnen abläuft, soll über eine Konfiguration geregelt werden.

Werden Tarife im System gefunden, folgt eine Bestimmung aller variablen Tarifbestandteile. Für diese wird der Kunde aufgefordert Werte festzulegen. Variable Tarifbestandteile können der Verbrauch aber auch die Zahlungsweise oder eine Tariffaufzeit sein.

Anschließend werden die Eingaben auf geltende Beschränkungen geprüft. Dies kann beim Verbrauch eine Minimal- oder Maximalangabe sein. Im Fehlerfall wird der Kunde darüber informiert und muss die Eingaben korrigieren.

Bei positivem Prüfungsergebnis werden für jeden Tarif die zutreffenden Tarifbestandteile zur Berechnung des Preises herangezogen. Abschließend erhält der Kunde eine Liste der verfügbaren Tarife mit den jeweilig geltenden Preisen. Sollte er die variablen Tarifbestandteile noch einmal ändern wollen, springt der Prozess zurück zur Eingabemaske. Anderenfalls wird der Tarifrechner an dieser Stelle beendet.

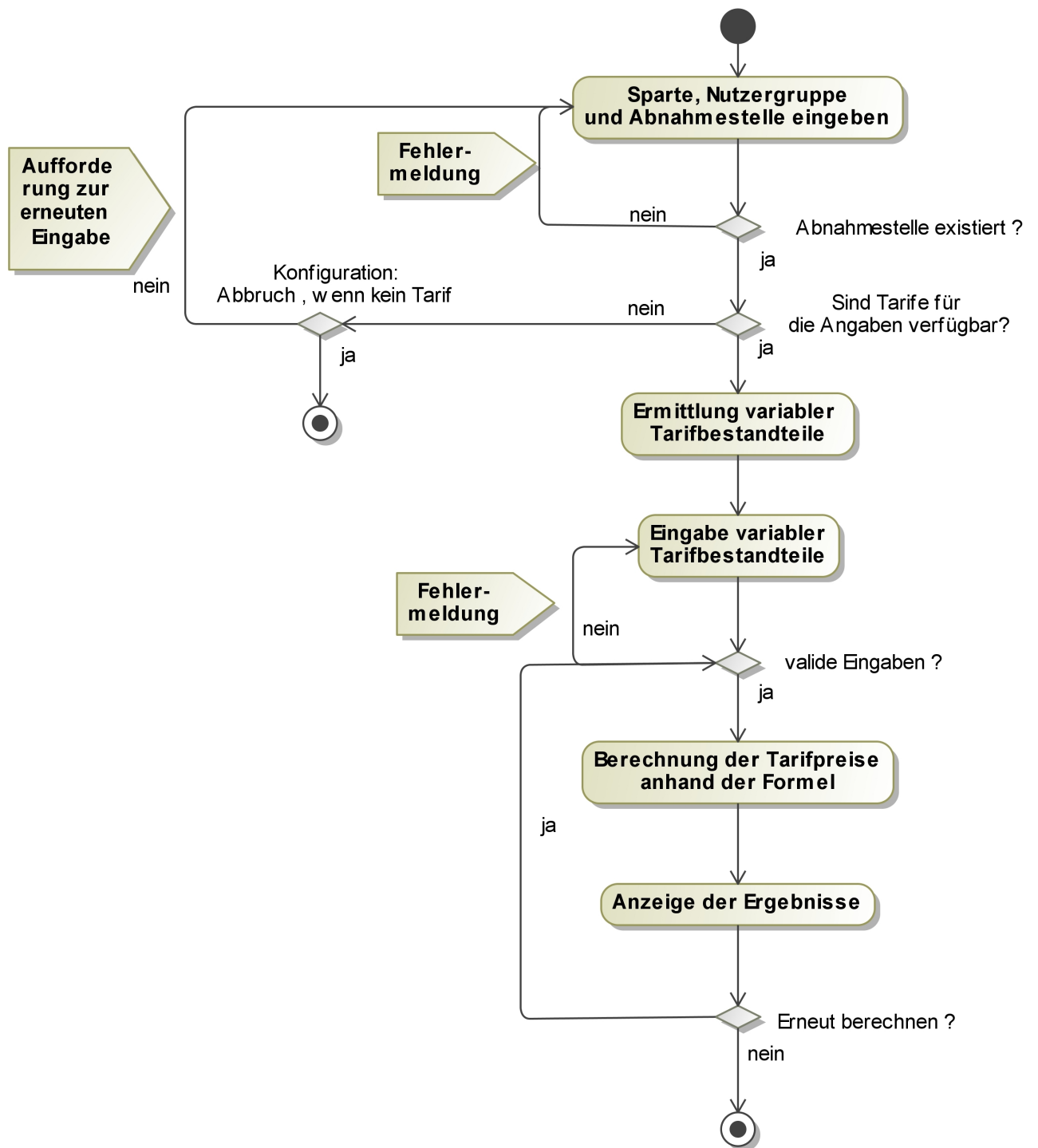


Abbildung 4.11.: Aktivitätsdiagramm Tarifrechner

4.3.3. Produktkonfigurator

Die derzeitig existierende Komplexität der Produkterstellung bzw. -erweiterung soll wesentlich vereinfacht und verkürzt werden. Dazu sollen einzelne Prozessschritte in komplett eigenständige Prozesse ausgelagert werden. Denn in der Regel wird die Routine zur Produktänderung nur wegen einer Namensänderung oder einer Preisanpassung aufgerufen. Dazu muss bis jetzt eine Vielzahl von Prozessschritten durchlaufen werden, ohne eine Eingabe auf diesen Seiten zu tätigen. Ein Abschluss der Bearbeitung zu einem früheren Zeitpunkt ist nicht möglich.

In der Abbildung 4.12 auf Seite 47 wird die Aufteilung des bisherigen Ablaufs in einzelne Teilprozesse deutlich. Dadurch können Daten zu Tarifen, Tarifbestandteilen, Tarifformeln und Liefergebieten separat gepflegt werden.

Der Prozess zur Tarifierstellung wird dabei eine Sonderrolle einnehmen, da er die einzeln angelegten Tarifelemente zu einem Produkt verknüpfen kann. Eine Beschreibung erfolgt im späteren Text.

Tarifteile sind die Elemente eines Tarifes, welche in einer Formel rechnerisch aneinandergereiht werden, um den Preis eines Produktes zu bestimmen. Im einfachsten Fall sind Grundpreis, Verbrauchspreis und Verbrauch Bestandteile eines Tarifes. PowerCommerce unterscheidet dabei zwischen zwei Typen: Nutzereingaben (Verbrauch) und Festpreise (Grundpreis, Verbrauchspreis).

Für Preisbestandteile wird im Produktkonfigurator die Eingabe der Währung und des entsprechenden Preiswertes nötig. Weiterhin wird festgelegt, auf welche Einheit sich diese Preisangabe bezieht. Beispielsweise kann sich der Grundpreis auf einen Betrag pro Monat beziehen, während der Verbrauchspreis eine Angabe pro Kilowattstunde ist. Bei Nutzereingaben ist lediglich die Bezugseinheit der Angabe zu wählen. Beim Verbrauch ist dies für Stromtarife meist Kilowattstunden.

Ein neuer Tarifbestandteilstyp wird die Tarifoption sein. Sie stellt eine Kombination aus Preisinformationen und Nutzereingaben dar. Das soll am Beispiel der Vertragslaufzeit näher erklärt werden:

Ein Energieversorgungsunternehmen bietet ein Produkt mit variabler Vertragslaufzeit an. Der Kunde hat die Wahl zwischen einer Vertragsdauer von drei, sechs oder zwölf Monaten. Um die lange Vertragsbindung des Kunden zu honorieren, bekommt der Verbraucher einen Preisnachlass, je länger die Dauer gewählt wird. Diese, nicht zwingende, Auswahl bezeichnen wir als Tarifoption.

Bei der Festlegung eines Tarifbestandteils als Tarifoption müssen demnach die einzelnen Optionswerte, sowie die zugehörigen Preisveränderungen angelegt werden.

Für alle Tarifbestandteile können zusätzlich Attributswerte bestimmt werden. Verbrauchsunter- und obergrenzen sind dabei die typischsten Anwendungsfälle.

4. Konzeption

Eine weitere Angabe zu den Bestandteilen ist die Übersetzung. Hier werden, ein dem Nutzer verständlicher Name und eine kurze Beschreibung hinterlegt. Diese dienen dem Endkunden als Eingabehilfe bei der Nutzung des Onlineservice.

In der Tarifformelverwaltung können mathematische Ausdrücke erstellt werden, um die Preisermittlung eines Produktes zu bestimmen. Dazu werden Tarifbestandteile anhand ihrer Namen mit arithmetischen Operationszeichen verknüpft. Eine einfache Formel wäre: „Ergebnis = Grundpreis + Verbrauch * Verbrauchspreis“.

Ein weiterer Anwendungsfall des Produktkonfigurators ist die Verwaltung eines Liefergebietes. Dazu können Postleitzahlen, Orte und Straßen zu einem Versorgungsgebiet zusammengefasst werden. Auch das Ausschließen einzelner Teile aus einem Areal ist möglich. Da meist mehrere Tarife ein und dasselbe Liefergebiet bedienen, muss dieses nur einmal angelegt werden, um es anschließend den entsprechenden Produkten zuzuordnen.

Da Tarifbestandteile, die Formel zur Berechnung, sowie das Liefergebiet für mehrere Produkte wiederverwendet werden können, wird der Produktkonfigurator nach einem Baukastenprinzip arbeiten. Es werden die Elemente zunächst einzeln angelegt oder verändert. Im Prozess 'Produkt erstellen' können die einzelnen Elemente zusammengefügt und zu einem Produkt verbunden werden. Dadurch muss beispielsweise eine Änderung an dem Liefergebiet eines Tarifes nur einmalig erfolgen, wenn dieses sich für alle im System hinterlegten Produkte ändert.

Die Produkterstellung erfolgt dabei nach folgenden Schritten (Abbildung 4.13, Seite 48).

1. Eingabe allgemeiner Tarifparameter, wie Name, Beschreibung, Bild. Festlegung von Laufzeiten und Kündigungsfristen. Einstellung, ab wann ein Tarif gültig ist (abschließbar) und ab wann er sichtbar ist. Weiterhin werden die Notwendigkeit von Unterschriften, der Zählertyp sowie die Sparte festgelegt.
2. Prüfung der Eingaben auf Plausibilität und anschließende Speicherung des Tarifes.
3. Wenn bereits Tarifbestandteile im Programm hinterlegt sind, können diese dem Tarif nun zugeordnet werden. Selbiges gilt für die Tarifformel und das Liefergebiet.
4. Wenn einem Tarif alle notwendigen Bestandteile zugeordnet wurden, kann der Tarif freigeschaltet werden und wird dem Endkunden des EVU damit zugänglich.

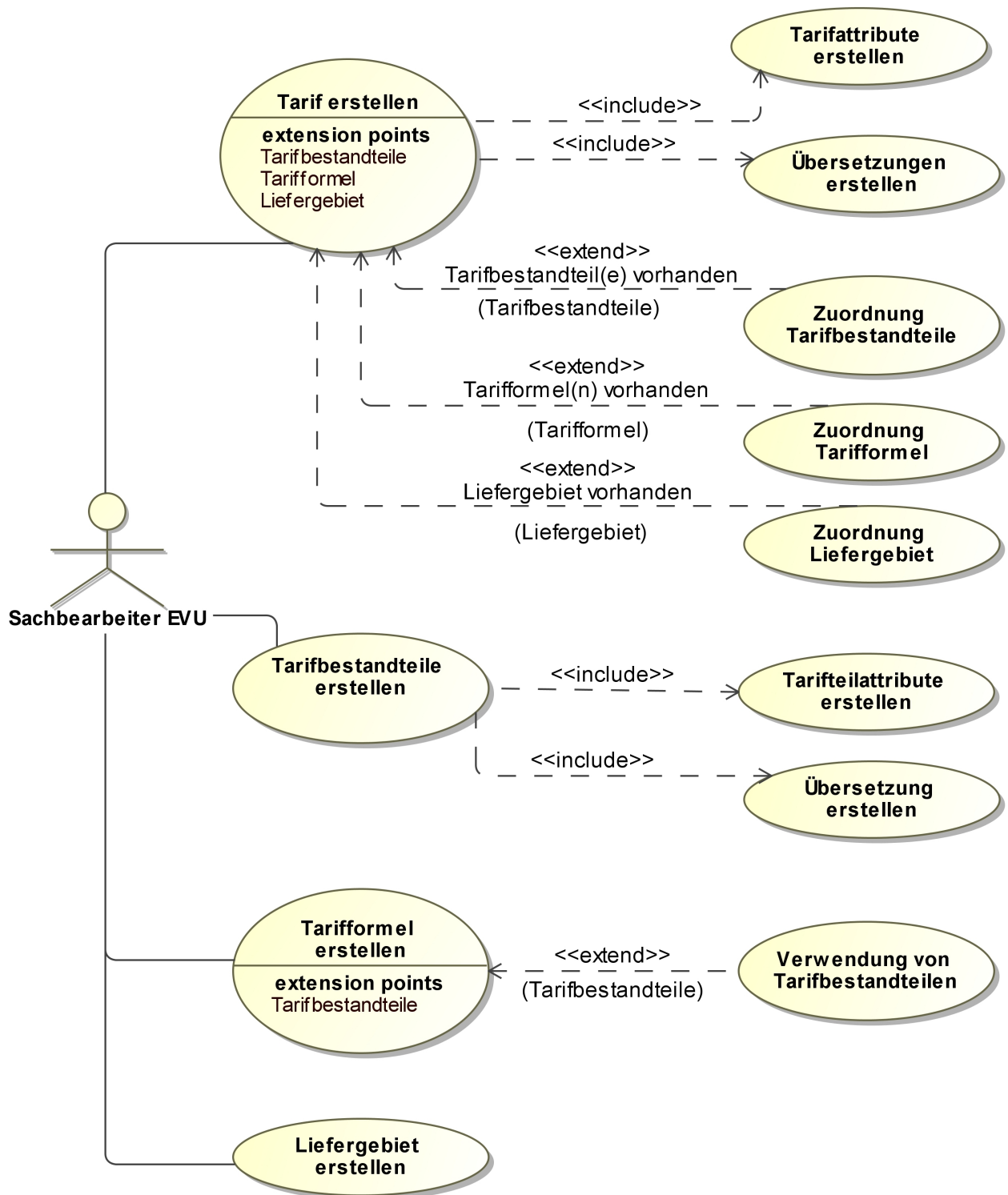


Abbildung 4.12.: Anwendungsfalldiagramm Produktkonfigurator

4. Konzeption

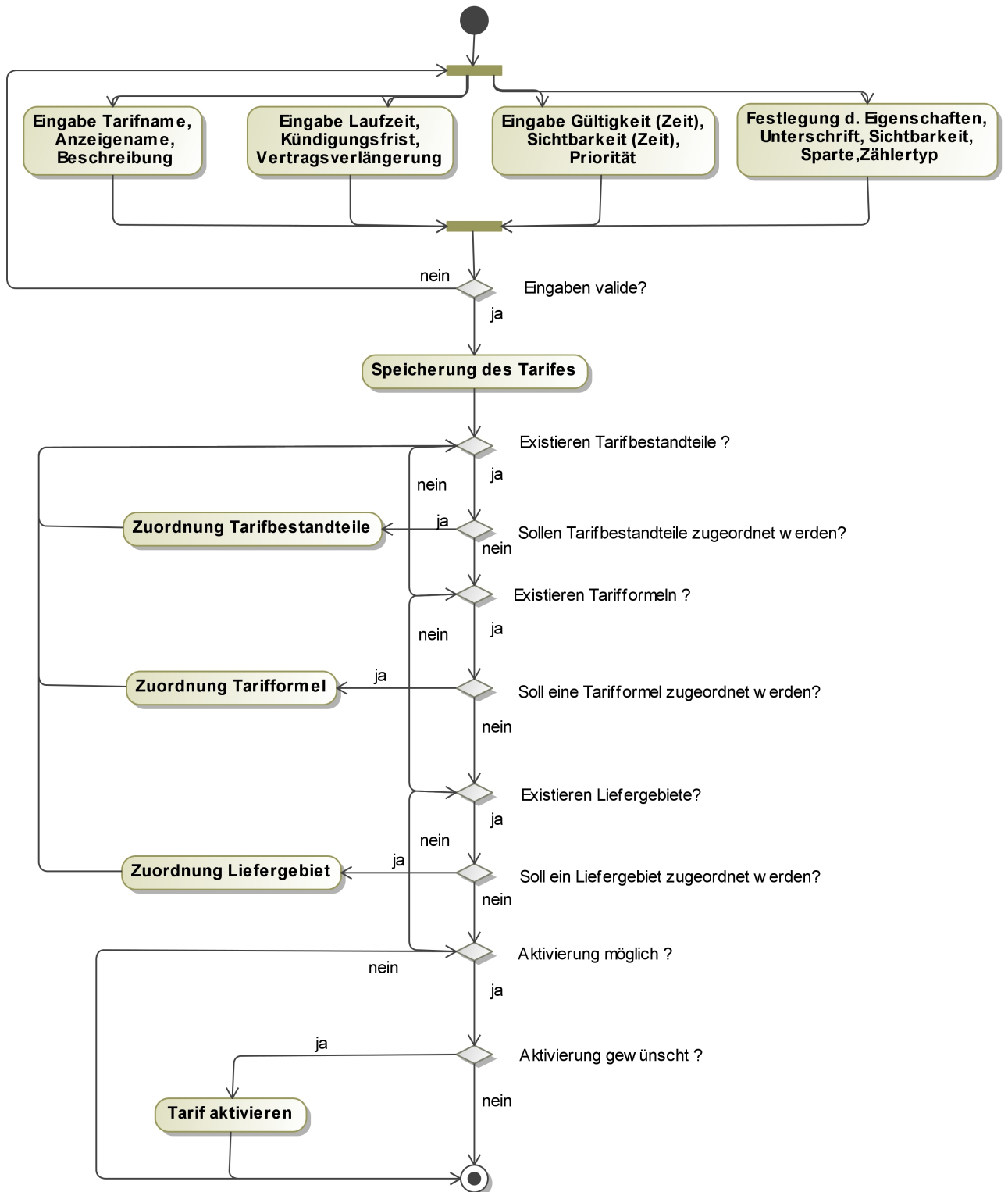


Abbildung 4.13.: Aktivitätsdiagramm Produkt erstellen

5. Implementierung

5.1. Tarifmodell

Zur Realisierung der Tarifooptionen wird das bestehende Datenmodell um die Tabellen PC_TariffOptionValue und PC_TariffOptionValueI18N erweitert.

Sie enthalten im Wesentlichen die zur Auswahl stehenden Optionsalternativen, sowie deren Übersetzungen. Die Tabelle PC_TariffOptionValue enthält folgende Spalten:

pcid Primärschlüssel

optionValue enthält den eigentlichen Optionswert, muss pro Tarifooption eindeutig sein

isDefault Dieses Feld gibt an, ob der Optionswert als Standardwert für die Tarifooption angenommen wird. Wird keine Auswahl zu der Tarifooption getroffen, wird der Standardwert zur Berechnung des Tarifpreises eingestellt.

partID Enthält die pcID des Tarifbestandteils.

creationDate Zeitstempel der Erstellung des Datensatzes

lastModified Zeitstempel der letzten Änderung des Datensatzes

status Status des Datensatzes

Folgende Spaltendefinition ist für die Tabelle PC_TariffOptionValueI18N vorgesehen:

pcid Primärschlüssel

optionValueI18N enthält die Übersetzung des Optionswertes für die angegebene Sprache

i18nLocale Die Sprache des Datensatzes

optionValueID Enthält die pcID des OptionsWertes.

creationDate Zeitstempel der Erstellung des Datensatzes

lastModified Zeitstempel der letzten Änderung des Datensatzes

status Status des Datensatzes

Die bereits vorhandenen Tabellen PC_TariffPart und PC_Price wurden angepasst, um die Tarifoptionen realisieren zu können.

Bisher enthielt die PC_TariffPart eine Spalte *userValue* vom Typ Boolean. Diese zeigte an, ob es sich bei diesem Tarifbestandteil um eine Nutzereingabe oder einen Festpreis handelte. Bei der Umstellung des Datenmodells, wurde diese Spalte durch eine Spalte des Datentyps Integer *tariffPartType* ersetzt. Dadurch können in Zukunft problemlos neue Bestandteilstypen definiert werden, ohne eine Anpassung der Datenstrukturen vornehmen zu müssen. Weiterhin bietet sich damit die Möglichkeit, Bitverknüpfungen abzubilden, um möglicherweise auftretende Kombinationen von Typinformationen zu realisieren.

Legte man bisher einen Festpreisbestandteil im Datenbestand an, so wurden die dazu nötigen Preisinformationen in der PC_Price hinterlegt. Um diese Verknüpfung zu realisieren, wurde der Primärschlüssel des Tarifbestandteils als Fremdschlüssel in der Preistabelle unter *tariffPartID* gespeichert. Durch die Tarifoptionen ist diese Spaltenbezeichnung nicht mehr korrekt. Denn für Tarifoption werden die Preise nicht mit dem Tarifbestandteil, sondern mit seinen darunterliegenden Tarifoptionswerten verknüpft. Zu diesem Zweck trägt die Spalte nun den Namen *ownerID*.

Bei den Betrachtungen zu einigen Tarifen der EVUs ist aufgefallen, dass bei Auswahl eines Tarifptionswertes sich sowohl Arbeits- als auch Grundpreis verändern sollen. Das bedeutet, dass es nicht ausreichend ist, pro Optionswert einen Preis hinterlegen zu können. Es muss nunmehr möglich sein, für jeden Festpreistarifbestandteil eine Veränderung des Preises vornehmen zu können. Dazu wurde die Tabelle PC_Price um das Feld *modifyPart* erweitert. Darin wird der technische Name des Tarifbestandteils gespeichert, für welchen die Preisanpassung stattfinden soll. Eine weitere Neuerung für die Preisinformation ist die Möglichkeit, prozentuale Angaben machen zu können. Es ist durchaus geläufig, bei Auswahl einer bestimmten Tarifoption dem Kunden einen Rabatt von x-Prozent auf seinen Grund-, Arbeits- oder Jahresgesamtpreis zu gewähren. Um zu kennzeichnen, ob es sich bei einem Preis um eine Prozentangabe handelt, wurde die Spalte *isPercentage* angelegt. Da man davon ausgehen kann, dass als Preis lediglich Prozentangaben oder konkrete Angaben existieren, wurde dafür der Datentyp Boolean gewählt.

Am Beispiel einer Festpreisgarantie¹ soll das System näher erläutert werden: Ein Tarif mit einer variablen Preisgarantie wird vom EVU angeboten. Dabei soll der Kunde zwischen drei, sechs und zwölf Monaten wählen können. Je länger er eine Preiserhöhung

¹Durch eine Festpreisgarantie verpflichtet sich ein EVU, den Preis innerhalb des angegebenen Zeitraums nicht zu erhöhen.

5. Implementierung

unterbinden möchte, desto höher soll sein eigentlicher Arbeitspreis sein.

Im Datenbestand würden dazu folgende Objekte angelegt werden:

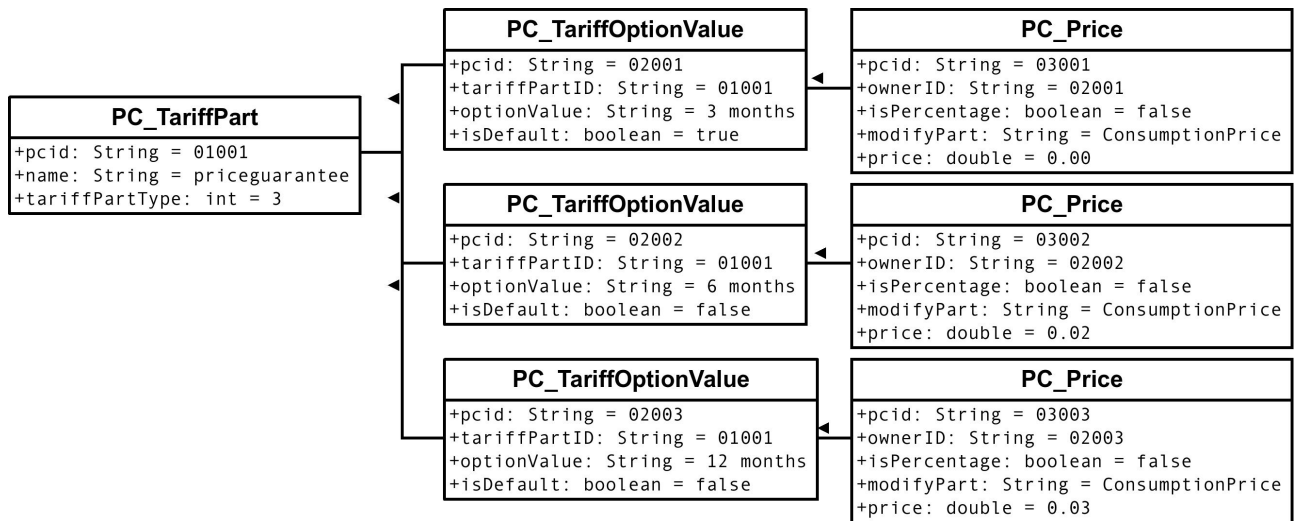


Abbildung 5.1.: Objekte einer Tarifoption 'Preisgarantie'

In der Abbildung 5.1 auf der Seite 51 wurde auf die Tabellen zur Speicherung der Übersetzungsdaten verzichtet, da sie für das Verständnis der allgemeinen Zusammenhänge der Tarifoptionen nicht notwendig sind.

5.2. Tarifrechner

Der bestehende Tarifrechner ermittelt in den ersten beiden Schritten zunächst die relevanten Tarife für den Nutzer. Dazu werden Kundentyp, Sparte und Postleitzahl oder Ort abgefragt. Sollten für dieses Liefergebiet Tarife mit unterschiedlichen Zählertypen zur Auswahl stehen, wird der Nutzer anschließend aufgefordert, den bei ihm verbauten Zählertyp anzugeben.

Die dritte Seite des Produktrechners ist für die Eingabe der Tarifbestandteile vorgesehen, die als Nutzereingaben deklariert wurden. Dafür werden aus allen noch zur Auswahl stehenden Tarifen die Nutzereingaben geholt und Eingabefelder generiert. Wenn Tarifbestandteile den selben Namen haben, wird davon ausgegangen, dass es sich inhaltlich um den selben Bestandteil handelt und nur ein Eingabefeld angezeigt.

Diese Seite wurde erweitert, um die Tarifoptionen dem Nutzer zur Auswahl zu stellen. Dabei ist das Vorgehen zur Anzeige der Auswahlmöglichkeiten ähnlich. Aus allen möglichen Tarifen werden die Tarifoptionen geholt. Wenn Optionen den gleichen technischen Namen haben, wird davon ausgegangen, dass sie inhaltlich das Selbe repräsentieren. Die Auswahlalternativen werden für alle Tarifoptionen des selben Namens zusammengefasst. So kann es sein, dass ein Tarif A eine Preisgarantie mit drei und sechs Monaten anbietet und ein Tarif B sechs und zwölf Monate. Im Tarifrechner wird dafür aber nur ein Block *Preisgarantie* erstellt und die drei Auswahlmöglichkeiten drei, sechs und zwölf Monate angeboten.

The screenshot displays a web form titled 'Eingabewerte' (Input Values) and 'Tarifoptionen' (Tariff Options). In the 'Eingabewerte' section, there is a label 'Stromverbrauch:' followed by a text input field containing '3000' and a unit selector 'Jahr'. The 'Tarifoptionen' section contains two groups of radio buttons. The first group, 'Zahlungsart:', has four options: 'keine Angabe' (selected), 'Barzahler', 'Überweisung', and 'Bankeinzug'. The second group, 'Preisgarantie:', has four options: 'keine Angabe' (selected), '3 Monate Preisgarantie', '6 Monate Preisgarantie', and '12 Monate Preisgarantie'. At the bottom of the form, there are two buttons: 'Zurück' (Back) and 'Berechnen' (Calculate).

Abbildung 5.2.: Nutzereingaben und Tarifoptionen im Produktrechner

5. Implementierung

Anschließend erfolgt die eigentliche Berechnung der Tarifpreise, für die vom Nutzer gewählten Parameter. Dazu müssen die Variablen der Tarifformel durch die Angaben des Produktrechners ersetzt werden. Die Eingabemaske für die Nutzereingaben sieht für jede Tarifoption die Auswahl *-keine Angabe-* vor. Wenn der Interessent diese Auswahl trifft, signalisiert er, dass er sich diesbezüglich noch nicht festlegen möchte. In diesem Fall wird die Standardauswahl für die Tarifoption getroffen.

In Listing A.1 auf Seite 66 ist ein Quellcodeausschnitt zur Bestimmung des Variablenwertes für die Tarifoptionen zu sehen.

Die eigentliche Berechnung des Tarifpreises erfolgt im Tariffmanager. Bisher wurde beim Anlegen oder Bearbeiten eines Tarifes dessen Formel in eine Javaklasse umgewandelt und der dazugehörige Bytecode wurde in die Datenbank geschrieben. Bei jeder Verwendung der Formel wurde ein Objekt dieser Klasse erzeugt. Die zur Berechnung notwendigen Parameter wurden übergeben und als Rückgabewert erhielt man das Ergebnis. Da dieses Vorgehen eine Fehlersuche innerhalb der Formel nahezu unmöglich macht und durch den Formelstring, sowie den generierten Bytecode eine redundante Datenhaltung vorliegt, wurde in der Neuentwicklung davon abgesehen. Um die Tarifpreise zu berechnen wird mit Groovy² gearbeitet. Die Tarifformel wird in ein Groovyskript umgewandelt, welchem anschließend die vom Nutzer eingegebenen oder vom System bestimmten Variablenwerte übergeben werden. Dieses Skript wird innerhalb der Groovy-Shell ausgeführt und liefert damit das berechnete Ergebnis zurück.

Ein Codeausschnitt zur Berechnung des Tarifpreises ist auf Seite 68 in Listing A.2 zu sehen. Zunächst (Zeilen 5-8) wird geprüft, ob die Berechnung des Tarifpreises für diesen Tarif ignoriert werden soll. Diese Einstellung kann beim Anlegen bzw. Bearbeiten eines Produktes im Backoffice vorgenommen werden.

Auf der Zeile 14 ist die Initialisierung des Groovy-Bindings zu sehen. Dieses nimmt die an das Skript (die Tarifformel) zu übergebenden Variablen auf.

Die vom Nutzer eingegebenen Tarifbestandteile, wie Verbrauch, sowie die festen Preisbestandteile, wie Grundpreis, werden in den Zeilen 17-20 bzw. 28-33 als Variable an das Binding übergeben. Die Preise der festen Bestandteile werden zwischengespeichert, damit diese durch die Tarifoptionen anschließend manipuliert und anteilig zum aktuellen Preis addiert werden können.

Die Verarbeitung der Tarifoptionen ist zunächst etwas aufwendiger, da durch sie andere Tarifbestandteile oder das Gesamtergebnis beeinflusst werden kann (Zeilen 36 - 100). Im ersten Schritt wird der gewählte Optionswert bestimmt, oder falls dieser nicht vorhanden ist, der Standardwert dieser Tarifoption geladen.

Zur weiteren Berechnung sind nur die Preise eines Tarifoptionswertes interessant, die prozentuale Angaben enthalten oder die nutzerspezifische Währung haben. Dabei muss

²Groovy ist eine dynamische Programmier- und Skriptsprache. Sie eignet sich hervorragend zur Integration in Java und erweitert es damit um nützliche und komfortable Features.

5. Implementierung

noch unterschieden werden, ob ein Preis das Gesamtergebnis beeinflusst und damit nach der eigentlichen Preisberechnung noch dazuaddiert wird (Zeile 95), oder aber einen anderen festen Tarifbestandteil beeinflusst.

Manipuliert ein Preis einen festen Tarifbestandteilspreis prozentual, so wird der entsprechende Prozentwert aus dem zwischengespeicherten Originalwert gebildet und zum aktuell im Binding hinterlegten Variablenwert addiert (Zeilen 72-73).

Bei direkter Beeinflussung eines Preises wird der aktuelle Variablenwert des Binding um den Preiseinfluss erweitert (Zeilen 88-89). Die Zeilen 107-126 zeigen die eigentliche Berechnung der Formel mit Hilfe von Groovy. Dazu wird die Formel in ein Groovy-Shell-Skript umgewandelt, welchem anschließend die im Binding gesammelten Variablenwerte übergeben werden. Durch die Ausführung des Skripts erfolgt die Evaluierung der Formel und die Berechnung des Tarifpreises. Dieser wird als Ergebnis von der Groovy-Shell zurückgegeben.

Tarifoptionen, die das Gesamtergebnis beeinflussen, werden anschließend auf dieses addiert. Dabei wird analog zur Manipulation anderer Tarifbestandteile (wie oben beschrieben) verfahren (Zeilen 129 - 145).

Das Endergebnis der Berechnung wird als Rückgabewert der Methode für jeden Tarif auf der Ergebnisseite des Tarifrechners ausgegeben.

5.3. Produktkonfigurator

Um den Produktkonfigurator für Tarifoptionen vorzubereiten, wurde die Eingabemaske der Tarifbestandteile erweitert (Abbildung 5.3 auf Seite 55).

Abbildung 5.3.: Tarifoptionen im Produktkonfigurator pflegen

Die Auswahl vorhandener Ids wurde um die Tarifoptionen erweitert. Da im Produktrechner inhaltlich gleiche Tarifoptionen anhand ihrer Id erkannt werden, sollte dieser Name gleich sein. Ist er verschieden, wird die Tarifoption als unterschiedlich erkannt und es wird eine neue Auswahl der möglichen Werte angeboten³.

Die zuvor über Radiobuttons realisierte Auswahl des Tarifparttyps erfolgt nun über eine Auswahlliste. Damit wird ein problemloses Hinzufügen weiterer Typen möglich, ohne an der Optik etwas zu verändern. Die Auswahlliste wurde um eine JavaScript-Funktion beim Klicken auf die einzelnen Auswahlen erweitert, damit nur die Eingabefelder angezeigt werden, die für den gewählten Tarifparttyp nötig sind.

Wird der Type *tarifoption* gewählt, erscheint noch eine weitere Auswahlliste: der Optionstyp. Einige Tarifoptionen, wie zum Beispiel die Zahlungsart oder der Rechnungstyp, können Auswirkungen auf den Ablauf einiger Geschäftsprozesse haben. Denn wenn ein

³siehe dazu Abbildung 5.2 auf Seite 52

Kunde sich für die Option *Online-Rechnung* entscheidet, sollte beim Vertragsabschluss nicht die abweichende Rechnungsanschrift abgefragt werden. Um auf diese Besonderheiten in den einzelnen Geschäftsprozessen reagieren zu können, werden einige Optionstypen und die dazugehörigen Werte vordefiniert. Wählt der Sachbearbeiter einen solchen Optionstyp aus, kann er die Optionswerte nicht frei eingeben. Es wird ihm eine Auswahlliste möglicher Werte angeboten.

Für die Pflege der Tarifoptionswerte wurde ein neuer Block entworfen - einen Templateausschnitt ist in Listing A.3 auf Seite 72 zu sehen. Im linken Bereich werden die einzelnen, schon angelegten Tarifoptionswerte aufgelistet und es können neue Werte angelegt werden. Dazu dienen ein Eingabefeld und eine Auswahlliste. In das Eingabefeld können neue Tarifoptionswerte frei eingetragen werden. Wenn eine Tarifoption mit der gleichen ID schon für einen anderen Tarif erstellt wurde, können hier die Tarifoptionswerte in der Auswahlliste gewählt werden. Im Produktrechner werden gleiche Optionswerte anhand ihres Namens erkannt, um nur eine Auswahl für inhaltlich identische Werte zu erstellen. Damit ein Verschreiben beim Erstellen der Optionswerte verhindert werden kann, enthält die Auswahlliste bereits vorhandene Optionswerte anderer Tarifooptionen mit der gleichen ID.

Sobald ein angelegter Tarifoptionswert angewählt wird, erscheinen im rechten Teil des Konfigurationsblocks weitere Einstellungsmöglichkeiten. Hier kann ein Optionswert wieder gelöscht oder als Standardwert definiert werden. Der Standardwert einer Tarifoption wird immer dann gewählt, wenn der Nutzer im Produktrechner *keine Angabe* markiert hat. Weiterhin ist dieser Wert beim Vertragsabschluss oder dem Produktwechsel vorselektiert.

Unterhalb dieser Angaben, kann der Bearbeiter Rabatte oder Aufschläge für den gewählten Optionswert anlegen. Beim Klicken auf *Preis hinzufügen* wird eine neue Zeile zur Preisbestimmung erzeugt. Für jeden Preis kann der Bezugspunkt, der Wert und die Einheit festgelegt werden. Als Bezugspunkte stehen dem Benutzer die festen Tarifbestandteile, sowie der Eintrag *Gesamtpreis* zur Auswahl. Manipuliert eine Tarifoption den Gesamtpreis, wird sie nach der eigentlichen Berechnung des Tarifpreises auf des errechnete Ergebnis angewendet. Bezieht sie sich auf einen anderen Tarifpart, wird der Aufschlag bzw. der Rabatt auf den Preis des gewählten Bezugspunktes vor der Berechnung des Preises angewendet⁴.

Der Wert der Preisangabe kann sowohl negativ als auch positiv erfolgen. Negative Angaben werden als Rabatt, positive als Aufschlag, interpretiert. Bei der Einheit des Preises kann zwischen *Euro* und *Prozent* gewählt werden. Prozentuale Preisangaben beziehen sich immer auf den ursprünglich eingestellten oder berechneten Wert des gewählten Bezugspunktes.

Bereits angelegte Preisinformationen können über die Schaltfläche *löschen* wieder ent-

⁴Details dazu im Kapitel 5.2

5. Implementierung

fernt werden.

Das Erstellen oder Bearbeiten einer Tarifoption kann über Klicken von *Tarifpart hinzufügen* bzw. *Tarifpart ändern* abgeschlossen werden.

In den weiteren Schritten des Tarifkonfigurators wird die Formel und das Liefergebiet des Tarifes abgefragt. Für die Tarifoptionen muss die Formel nicht extra angepasst werden. Wie Abbildung 5.4 auf Seite 57 zeigt, werden im oberen Bereich der Seite die zuvor angelegten Tarifbestandteile aufgelistet. Tarifoptionen werden, wie die Nutzereingaben, gesondert gekennzeichnet.

Eingegebene Tarifparts	
Eingegebene Tarifparts	Vordefinierte Attribute (Min / Max / optional)
AchievementPrice	
ConsumptionPrice	
PowerConsumption (Nutzereingabe)	100 / 10000 / false
Priceguarantee (Tarifoption)	

Verfügbare Tarifformeln	
Power_OneMeter	Anzeigen Übernehmen
Power_TwoMeter	Anzeigen

Details

Hilfe ?

Tarifformelname:

Power_OneMeter

Tarifformel:

result = AchievementPrice + (PowerConsumption * ConsumptionPrice);

Zurück

Weiter

Abbildung 5.4.: Tarifformel im Produktkonfigurator pflegen

Im unteren Bereich werden im System hinterlegte Tarifformeln aufgeführt und können angesehen oder übernommen werden. Eine Formel kann übernommen werden, sobald alle in ihr vorkommenden Tarifbestandteile auch für den gewählten Tarif gepflegt worden sind.

Im Eingabebereich am Ende der Seite kann der Formelname und die Zeichenkette der Tarifformel bearbeitet werden. Eine Tarifformel darf nur für diesen Tarif angelegte Tarifbestandteile als Variablen enthalten. Die Tarifoptionen müssen nicht in die Formel mit aufgenommen werden, da die Bezugspunkte auf der vorherigen Seite festgelegt wurden und sie somit die hier angegebenen Tarifbestandteile direkt oder indirekt über das Gesamtergebnis verändern.

Im letzten Prozessschritt des Produktkonfigurators werden die Anzeigetexte des Tari-

5. Implementierung

fes sowie der Tarifbestandteile gepflegt. Die hier eingegebenen Texte werden dem Kunden angezeigt. Für die Tarifoptionen stehen neben der Bezeichnung der Tarifoptionen noch Felder für die Übersetzung jedes Optionswertes zur Verfügung.

Nach erfolgreicher Eingabe der Texte ist der Änderungsprozess abgeschlossen und der Produktkonfigurator wird beendet.

5.4. Schnittstellen

Die vorhandene XML-Schnittstelle zum Tarifimport und -export wurde um die Funktionalität der Tarifoptionen erweitert. Für beide Anwendungsfälle werden die Tarife und ihre Daten in einer DOM-Struktur in eine Datei abgelegt.

Ein Unterelement der Tarife sind dabei die Tarifbestandteile (*tariffPart*, siehe dazu Listing A.4 auf Seite 76). Zusätzlich zu den bisherigen Daten werden nun noch die Tarifoptionen und ihre entsprechenden Unterelemente in den DOM-Baum mit aufgenommen. In den Zeilen 65 bis 133 des Listing ist die Realisierung der Tarifoption *Preisgarantie*, die schon einmal als Beispiel genommen wurde, dargestellt.

Listing 5.1: Ausschnitt aus Tarifimport mit Tarifoption

```
...
<tariffPart>
  <name>priceguarantee</name>
  <i18n>
    <lang>
      <locale>de_DE</locale>
      <displayName>Preisgarantie</displayName>
      <shortName/>
      <description/>
    
```

Bisweilen wird beim Tarifimport ein DOM-Parser verwendet, welcher das zu importierende Dokument zunächst einlädt und in eine Baumstruktur umsetzt. Diese wird anschließend Knoten für Knoten durchgeklettert und die entsprechenden Elemente des Knotens werden in der Datenstruktur von PowerCommerce angelegt. Beim Importieren eines Tarifes wird weiterhin geprüft, ob dieser bereits im Datenbestand vorliegt. Ist das der Fall, werden die vorhandenen Daten durch die neuen aktualisiert. Ist der Tarif noch nicht angelegt worden, so werden alle Elemente neu in der Datenbank erstellt.

Beim Exportieren wird über alle im Datenbestand vorhandenen Tarife und deren Unterelemente iteriert und deren Daten in einem Pufferspeicher zwischengehalten. Dieser wird anschließend in eine Datei geschrieben, welche dem Sachbearbeiter zum Herunterladen angeboten wird.

6. Ergebnis

6.1. Zusammenfassung

Im Rahmen dieser Diplomarbeit sollte für die E-Business-Applikation PowerCommerce eine Überarbeitung des Tarifmodells konzipiert und implementiert werden. Durch die zunehmende Kreativität bei der Tarifgestaltung und die kommende Einführung von SmartMetering war es nötig, Anpassungen am bestehenden System zur Datenverwaltung der Tarife vorzunehmen und die Berechnung der Tarifpreise anzupassen.

Ziel war es, eine Datenstruktur zu schaffen, die bisher aufgetretene Tarifkonstrukte abbildet und für mögliche Neuentwicklung vorbereitet ist. Die Gestaltung der neuen Datenhaltungsebene sollte so flexibel wie möglich sein, damit große Anpassungen für spätere Neuentwicklungen vermieden werden können.

Um für den Kunden eine transparente Preisinformation darstellen zu können, sollte der Tarifrechner an die neuen Tarifentwicklungen angepasst werden. Die Instrumentalisierung verschiedenster Tarifbestandteile sollte dabei in Form von Tarifoptionen realisiert werden.

Da sich PowerCommerce derzeit in einem Versionwechsel befindet, der einen Austausch des Frameworks und einen enormen Neuentwicklungsaufwand mit sich führt, war es nicht möglich alle gewünschten Änderungen umzusetzen.

In der Konzeption wurden sämtliche Neuentwicklungen für das Tarifmodell erdacht und beschrieben. Dadurch kann bei der Implementierung der nächsten PowerCommerce-Version die Datenstruktur grundlegend neu umgesetzt werden.

Die geforderte Flexibilität wurde über eine Modularisierung der einzelnen Tarifbestandteile realisiert. Tarifformel, Liefergebiet und Tarifbestandteile (Preise, Nutzereingaben und Tarifoptionen) werden nun als eigenständige Objekte gepflegt und über Aggregationstabellen zu einem Tarif zusammengefügt. Dadurch können die einzelnen Elemente jederzeit wiederverwendet und global verwaltet werden.

Die Prozesse *Tarifrechner* und *Produktkonfigurator* wurden konzeptionell überarbeitet und an die neue Datenhaltung angepasst. Die entscheidenden Änderungen werden dabei am Produktkonfigurator stattfinden. Dieser Prozess soll sich in mehrere Teilprozesse gliedern, um so das modularisierte Datenmodell unterstützen zu können. Weiterhin ist

6. Ergebnis

dadurch eine einfachere Handhabung bei der Bearbeitung und Erstellung von Tarifen gewährleistet.

In der eigentlichen Implementierung für diese Diplomarbeit wurden einige der konzipierten Änderungen in die bestehende PowerCommerce-Version übernommen. Da eine enorme Nachfrage durch die Energieversorgungsunternehmen bestand, wurden die Tarifoptionen in das aktuelle System eingebaut. Der Produktkonfigurator wurde angepasst, um die Tarifoptionen und ihre Werte erstellen und pflegen zu können. Für den Produktrechner wurden Anpassungen vorgenommen, um eine Berücksichtigung der Tarifoptionen zu ermöglichen.

Eine weitere Neuentwicklung ist die Interpretation der Tarifformel mit Hilfe der Skriptsprache *Groovy*. Es ist nun nicht mehr notwendig aus einer Tarifformel eine Klasse zu generieren, welche zur Berechnung der Preise herangezogen werden muss. Es wird lediglich der Formeltext als Skript evaluiert und das berechnete Ergebnis als Rückgabe ausgegeben.

Die bereits konzipierten aber noch nicht umgesetzten Änderungen werden in der nächsten Version von PowerCommerce implementiert.

6.2. Ausblick / Weiterentwicklung

Im Hinblick auf die Weiterentwicklung des Gegenstands dieser Diplomarbeit sind folgende Änderungen geplant, wünschenswert oder denkbar:

6.2.1. Produkte und Tarife - Export in das Abrechnungssystem

Die meisten Abrechnungssysteme unterscheiden zwischen Produkten und Tarifen. Dabei werden Produkte dem Kunden angeboten und haben verschiedenste Bestandteile, Beschreibungen und Bedingungen. Tarife sind die konkrete Ausprägung eines Produktes. Am Beispiel des Energiemarktes können zwei Kunden das Produkt *Ökostrom* haben, aber unterschiedliche Tarife beziehen. Durch die einzelnen Netzbetreiber fallen pro Abnahmestelle verschiedene Netznutzungsentgelte an. Diese können sich in den Ausprägungen von Tarifen eines Produktes wiederfinden.

PowerCommerce hat diese Unterscheidung bisher nicht. Beim Vertragsabschluss über PowerCommerce wird für den Tarif nur das eine, im Backoffice hinterlegte, Kürzel ins Abrechnungssystem exportiert. Durch die Tarifoptionen besteht nun aber die Möglichkeit verschiedene Ausprägungen eines Produktes nachzubilden. Es muss nun eine Möglichkeit geschaffen werden, für die verschiedenen Auswahlkombinationen unterschiedliche Exportkürzel für das Abrechnungssystem zu pflegen. Damit könnten die Tarife im PowerCommerce als Produkte gepflegt werden und durch die Vielzahl der Kombinationen der Optionen als Tarife exportiert werden.

6.2.2. Verbrauchshistorie mit Preisinformationen

PowerCommerce bietet derzeit dem Nutzer in der Verbrauchshistorie eine grafische Auflistung seiner im Datenbestand hinterlegten Verbrauchswerte an. Diese Funktion könnte in Zukunft um eine Preisinformation erweitert werden. Dazu müssten die Tarife und ihre Bestandteile historiesierbar sein. Anhand einer erstellten Änderungshierarchie kann zu jeder Zeit nachverfolgt werden, wann ein Tarif wie ausgesehen hat.

6.2.3. Zeitvariable Tarife

Um zeitvariable Tarife realisieren zu können, muss eine Möglichkeit geschaffen werden Zeitscheiben zu definieren. Als Zeitscheiben sind dabei komplexe Objekte zur Definition von Zeiträumen bzw. -abschnitten gemeint.

In die digitalen Stromzähler werden virtuelle Zeitscheiben programmiert, womit festge-

legt wird zu welcher Zeit und zu welchen Konditionen wieviel Energie verbraucht wurde. Dabei sind Einstellung von Feiertagen, Wochenenden oder Jahreszeiten denkbar. Diese Zeitscheiben müssen im PowerCommerce abgebildet werden, um zeitgenaue Preise verwalten zu können.

Sind Zeitscheiben im System hinterlegt, kann mit Hilfe der Tarifoptionen, ein zeitvariabler Tarif erstellt werden. Somit kann ein Arbeitspreis als Tarifoption angelgt und die verschiedenen Zeitscheiben als die Optionswerte hinterlegt werden. Bei der Berechnung eines Tarifpreises kann dann im System die derzeitig gültige Zeitscheibe ermittelt und so der dafür gültige Arbeitspreis berechnet werden.

6.2.4. Liefergebietsabhängige Preise

Für die nächste Version von PowerCommerce ist eine separate Pflege der Liefergebiete geplant. Dadurch können für einen Tarif Preise abhängig vom gewählten Liefergebiet eingestellt werden. Dazu wird ein Preis als Tarifoption gepflegt und erhält als Optionswerte verschiedene Teilgebiete des eigentlichen Liefergebietes.

Erfolgt ein Produktwechsel oder Vertragsabschluss kann anhand der Abnahmestelle das Liefergebiet und somit der Arbeitspreis bestimmt werden.

6.2.5. Einmalige Boni

Für die Anzeige von Produktpreisen im Tarifrechner, muss es die Möglichkeit geben, einmalige Bonuszahlung auszublenden, um dem Kunden eine transparente Preisinformation zu ermöglichen. Einmalige Zahlungen, wie *Wechselbonus*, sollen aber natürlich nicht als „Ja - Nein“- Auswahl beim Kunden abgefragt werden.

Daher sollte für Tarifoptionen die Einstellung geschaffen werden, wann eine Option vom Kunden selbst wählbar ist und wann sie ihm nicht zur Auswahl steht.

6.2.6. „Mindestens“ und „höchstens“ Filter

Für einige Tarifoptionen, wie Preisgarantie oder Vertragslaufzeit, ist eine Filterung im Produktrechner wünschenswert. Der Kunde könnte damit, die für ihn in Frage kommenden Tarife, schon vorselektieren und so seine Auswahl einschränken.

Da die Optionswerte vom Sachbearbeiter frei eingegeben werden können, ist eine dynamische Filterung schwierig. Es kann nicht mit Gewissheit gesagt werden, ob eine Preisgarantie mit dem Optionswert *3 Monate* oder *drei Monate* angelegt wurde.

6. *Ergebnis*

Es wäre denkbar, Filtereinstellungen pflegbar zu gestalten um damit festlegen zu können, welche Werte auf welchen Filter passen.

Literaturverzeichnis

- [EM0408] *Innovative Preis-Modelle in der Energiewirtschaft* - Artikel im Magazin „Energie & Management“ Ausgabe 04/2008
- [wBMWi] *Potenzziale der Informations- und Kommunikations-Technologien zur Optimierung der Energieversorgung und des Energieverbrauchs (eEnergy)* - wik-Consult, Fraunhofer ISI, Fraunhofer ISE, 21. Dezember 2006
- [YELLOW] <http://www.yellostrom.de> - Webseite der Yello Strom GmbH, zuletzt besucht: 26.05.2009
- [POCO] *Konzeption und Implementierung der Mandantenfähigkeit für die Softwarelösung PowerCommerce.* - Schuldt,Rico Diplomarbeit, Hochschule für Technik und Wirtschaft Dresden (FH), 2007.
- [VERI] <http://www.verivox.de> - Webseite der Verivox GmbH, zuletzt besucht: 29.05.2009
- [ToTa] <http://www.toptarif.de> - Webseite der Toptarif Internet GmbH, zuletzt besucht: 29.05.2009
- [SaLoLu] <http://www.energie-saarlorlux.com> - Webseite der Energie Saar-LorLux AG, zuletzt besucht: 29.05.2009
- [ENSO] <http://www.enso.de> - Webseite der ENSO Energie Sachsen Ost AG, zuletzt besucht: 29.05.2009

A. Anhang

Listing A.1: Bestimmung der Variablenwerte für Tarifoptionen

```
1  if (currentPart.getTariffPartType() == PCTariffConstants.  
    TARIFFPART_TYPE_OPTION)  
2  {  
3      if(options != null && !options.isEmpty() && options.get(  
        currentPart.getName()) != null)  
4      {  
5          String optionValue = (String) options.get(currentPart.getName())  
            ;  
6          if(optionValue != null)  
7          {  
8              PCTariffOptionValueTO optionValueTO =  
                getTariffOptionValueByValueTariffPartID(optionValue ,  
                    currentPart.getPcID());  
9              if(optionValueTO != null)  
10             {  
11                 Collection<PCPriceTO> prices = optionValueTO.getAllPrices();  
12                 if(prices != null)  
13                 {  
14                     for(PCPriceTO price : prices)  
15                     {  
16                         allVariables.put(  
17                             (price.getModifyPart() != null && price.getModifyPart().  
                                length() != 0)? currentPart.getName() + "." + price.  
                                    getModifyPart() : currentPart.getName(),  
18                             (price.getIsPercentage())? (100 + price.getPrice())/100  
                                : price.getPrice());  
19                     }  
20                 }  
21                 setDefaults = false;  
22             }  
23         }  
24     }  
25     //if no optionValue was set by the user, set defaultValues  
26     if(setDefaults)  
27     {
```

```

28 | PCTariffOptionValueTO defaultOptionValue = currentPart.
    | getDefaultOptionValue();
29 | if(defaultOptionValue != null)
30 | {
31 |     Collection<PCPriceTO> prices = (Collection<PCPriceTO>)
    |     defaultOptionValue.getAllPrices();
32 |     if(prices != null)
33 |     {
34 |         for(PCPriceTO price : prices)
35 |         {
36 |             allVariables.put(
37 |                 (price.getModifyPart() != null && price.getModifyPart().
    |                 length() != 0)? currentPart.getName() + "." + price.
    |                 getModifyPart() : currentPart.getName(),
38 |                 (price.getIsPercentage())? (100 + price.getPrice())/100 :
    |                 price.getPrice());
39 |         }
40 |     }
41 | }
42 | }
43 | }

```

Listing A.2: Berechnung des Tarifpreises mit Hilfe von Groovy

```

1 public Double calculateForTariff(
2 de.powercommerce.bean.tariff.PCTariffTO tariff, HashMap<String,
   Object> inputVariables, HashMap<String, String> options,
3 de.powercommerce.bean.currency.PCCurrencyTO currencyTO)
4 {
5     if(PCTariffConstants.TARIFFATTRIBUTEVALUE_TRUE.equals(tariff.
        getTariffAttributeValue(PCTariffConstants.
            TARIFFATTRIBUTE_IGNOREATPRODUCTCALCULATOR)))
6     {
7         return null;
8     }
9
10    Collection<PCTariffPartTO> fixParts =
        getTariffPartsByTariffPartTypeTariffID(PCTariffConstants.
            TARIFFPART_TYPE_FIXVALUE, tariff.getPcID());
11    Collection<PCTariffPartTO> optionParts =
        getTariffPartsByTariffPartTypeTariffID(PCTariffConstants.
            TARIFFPART_TYPE_OPTION, tariff.getPcID());
12
13    // create new Binding for all variables
14    Binding allVariables = new Binding();
15
16    //put all input variables in the complete list
17    for(String inputVariable : inputVariables.keySet())
18    {
19        allVariables.setVariable(inputVariable, inputVariables.get(
            inputVariable));
20    }
21
22    //save the default prices for fix tariffParts to be able to
        manipulate them by options later
23    HashMap<String, Double> originalFixPrices = new HashMap<String,
        Double>(fixParts.size());
24
25    Set<PCPriceTO> postProcessingPrices = new HashSet<PCPriceTO>();
26
27    //fixValues can be added to the binding directly
28    for(PCTariffPartTO tariffPartTO : fixParts)
29    {
30        Double price = getPriceByTariffPartAndCurrency(tariffPartTO,
            currencyTO);
31        allVariables.setVariable(tariffPartTO.getName(), price);
32        originalFixPrices.put(tariffPartTO.getName(), price);
33    }
34

```

```

35 //handling the tariffOptions
36 for(PCTariffPartTO tariffPartTO : optionParts)
37 {
38     //check if there is an optionValue choosen, instead use the
        default one
39     PCTariffOptionValueTO optionValueTO = null;
40     if(options != null && !options.isEmpty() && options.get(
        tariffPartTO.getName()) != null)
41     {
42         optionValueTO = getTariffOptionValueByValueTariffPartID(
            options.get(tariffPartTO.getName()), tariffPartTO.getPcID()
        );
43     }
44     else
45     {
46         optionValueTO = tariffPartTO.getDefaultOptionValue();
47     }
48
49     //tariffOptions can manipulate multiple parts so we have to
        check each price, which is percentage or has the right
        currency
50     Collection<PCPriceTO> prices = optionValueTO.getAllPrices();
51     if(prices != null)
52     {
53         for(PCPriceTO price : prices)
54         {
55             if(price.getIsPercentage() || price.getCurrencyID().equals(
                currencyTO.getPcID()))
56             {
57                 // if the price manipulates another part, so do it now,
                    else do it later
58                 if(price.getModifyPart() != null && price.getModifyPart().
                    trim().length() != 0)
59                 {
60                     if(price.getIsPercentage())
61                     {
62                         Double originalPrice = originalFixPrices.get(price.
                            getModifyPart());
63                         if(originalPrice == null)
64                         {
65                             log.error("the_tariff_" + tariff.getName() +
66                                 "_has_an_option_" + tariffPartTO.getName() +
67                                 "_which_manipulates_a_not_existing_tariffPart_" +
                                    price.getModifyPart());
68                             return null;
69                         }

```

```

70         else
71         {
72             double newPrice = originalPrice * price.getPrice() /
                100 + (Double) allVariables.getVariable(price.
                getModifyPart());
73             allVariables.setVariable(price.getModifyPart(),
                newPrice);
74         }
75     }
76     else
77     {
78         Double currentPrice = (Double) allVariables.
            getVariable(price.getModifyPart());
79         if(currentPrice == null)
80         {
81             log.error("the_tariff_" + tariff.getName() +
82             "_has_an_option_" + tariffPartTO.getName() +
83             "_which_manipulates_a_not_existing_tariffPart_" +
                price.getModifyPart());
84             return null;
85         }
86         else
87         {
88             double newPrice = price.getPrice() + (Double)
                allVariables.getVariable(price.getModifyPart());
89             allVariables.setVariable(price.getModifyPart(),
                newPrice);
90         }
91     }
92 }
93 else
94 {
95     postProcessingPrices.add(price);
96 }
97 }
98 }
99 }
100 }
101
102 /*
103  * the real calculation of the Formula
104  * 1. get the PCTariffFormula for the tariff
105  * 2. generate a script and add the variables
106  */
107 Double result = null;

```

```

108  if((tariff.getTariffFormulaID() != null) && (!tariff.
    getTariffFormulaID().equals("")))
109  {
110      try
111      {
112          PCTariffFormulaTO formula = getTariffFormulaByID(tariff.
            getTariffFormulaID());
113          Script script = groovyShell.parse(formula.getFormula());
114          script.setBinding(allVariables);
115          Object _result = script.run();
116          if(Double.class.isInstance(_result))
117          {
118              result = (Double) _result;
119          }
120      }
121      catch(Exception e)
122      {
123          result = null;
124          log.error("Error_calculating_Formula_for_tariff:" + tariff.
            getName());
125      }
126  }
127
128  //handling the tariffoptionprices referring the whole result
129  if(result != null && !postProcessingPrices.isEmpty())
130  {
131      double originalResult = new Double(result);
132      for(PCPriceTO price : postProcessingPrices)
133      {
134          if(price.getIsPercentage())
135          {
136              double newResult = originalResult * price.getPrice() / 100
                + result;
137              result = newResult;
138          }
139          else
140          {
141              double newResult = price.getPrice() + result;
142              result = newResult;
143          }
144      }
145  }
146
147  return result;
148  }

```

Listing A.3: Tarifoptionspflege im Produktkonfigurator

```

1 <div #if($constant.getConstant("de.powercommerce.bean.tariff.
    PCTariffConstants", "TARIFFPART_TYPE_OPTION") !=
    $changeProductForm.tariffPartType) style="display:none;#end>
2 <table class="default">
3 <tr>
4 <td class="label">Tariffoptionstyp:</td>
5 <td class="input">
6 <select name="optionType" class="long" onchange="refreshForm('
    changeProductForm');">
7 <option value="$constant.getConstant("de.powercommerce.bean.tariff.
    PCTariffConstants", "OPTIONTYPE_OTHER")" #if(!$changeProductForm.
    optionType || $changeProductForm.optionType.equals($constant.
    getConstant("de.powercommerce.bean.tariff.PCTariffConstants", "
    OPTIONTYPE_OTHER"))>selected="selected"#end>
8 Sonstige
9 </option>
10 <option value="$constant.getConstant("de.powercommerce.bean.tariff.
    PCTariffConstants", "OPTIONTYPE_PAYMENTMETHOD")" #if(
    $changeProductForm.optionType && $changeProductForm.optionType.
    equals($constant.getConstant("de.powercommerce.bean.tariff.
    PCTariffConstants", "OPTIONTYPE_PAYMENTMETHOD"))>selected="
    selected"#end>
11 Zahlungsart
12 </option>
13 </select>
14 </td>
15 </tr>
16 </table>
17 <table class="defaultOptions">
18 <tr>
19 <td class="longLabelOptions" style="background-color:#b6c1cd;_<
    strong>Tarifoptionswert</strong></td>
20 <td class="longInputOptions" style="background-color:#b6c1cd;_<
    strong>Tarifoptionspreis</strong></td>
21 </tr>
22
23 <tr>
24 <td class="longLabelOptions">&nbsp;</td>
25 <td class="longInputOptions">&nbsp;</td>
26 </tr>
27
28 ##if(!$!changeProductForm.nTariffPart && !$!changeProductForm.
    nTariffPart > -1)
29 ##set($tariffPartInfo = $changeProductForm.getTariffPart(
    $changeProductForm.nTariffPart))

```



```

30 #set($optionValues = $changeProductForm.optionValues)
31 #set($counter = 0)
32 #foreach($tariffOptionValue in $optionValues)
33 <tr>
34 <td class="longLabelOptions">
35 <input type="radio" name="selectedOptionValue" value="
    $tariffOptionValue.optionValue" onclick="refreshForm('
    changeProductForm');" #if($tariffOptionValue.optionValue ==
    $changeProductForm.selectedOptionValue)checked="checked"#end/>
36 $tariffOptionValue.optionValue ($tariffOptionValue.priority)
37 </td>
38 #if($counter == 0)
39 #set($size = $optionValues.size() + 2)
40 <td class="longInputOptions" rowspan="$size">
41 #foreach($tariffOptionValue in $optionValues)
42 #if($tariffOptionValue.optionValue == $changeProductForm.
    selectedOptionValue )
43 <input type="submit" name="btnDeleteOptionValue" class="minusList_
    btnInside" style="width:200px;" value="gew&auml;hlten_Optionswert
    _l&ouml;sch" />
44 <span style="padding-left:_20px;">
45 <input type="checkbox" name="OptionIsDefault($tariffOptionValue.
    optionValue)" value="true" #if($tariffOptionValue.isDefault)
    checked="checked"#end/>Standard
46 <input type="hidden" name="OptionIsDefault($tariffOptionValue.
    optionValue)" value="" />
47 </span>
48 <br style="clear:_both;" />
49 <br />
50 #foreach($price in $tariffOptionValue.prices)
51 #set($key = $tariffOptionValue.optionValue + $changeProductForm.
    getDelimiter() + $price.tempID)
52 <select class="long" name="PricesModifyPart($key)" style="width:_150
    px;">
53 <option value="" #if(!$price.modifyPart || $price.modifyPart.trim().
    length() == 0)selected="selected"#end>Gesamtpreis</option>
54 #foreach($tariffPart in $changeProductForm.tariffParts)
55 #if($constant.getConstant("de.powercommerce.bean.tariff.
    PCTariffConstants","TARIFFPART_TYPE_FIXVALUE") == $tariffPart.
    tariffPartType)
56 <option value="$tariffPart.tariffPartID" #if($price.modifyPart.
    equals($tariffPart.tariffPartID))selected="selected"#end>$!
    stringTool.txt2xml($tariffPart.tariffPartID)</option>
57 #end
58 #end
59 </select>

```

```

60 <input type="text" class="short" name="PricesPrice($key)" value="$!
    stringTool.txt2xml($price.priceString)" />
61 <select name="PricesIsPercentage($key)">
62 <option value="false" #if(!$price.isPercentage) selected="selected"##
    end>Euro</option>
63 <option value="true" #if($price.isPercentage) selected="selected"##
    end>%</option>
64 </select>
65 <input type="submit" name="BtnDeleteOptionValuePrice($price.tempID)"
    class="minusList_btnInside" style="float:none;_width:_62px;"
    value="l&ouml;schen" />
66 #end
67 <br />
68 <br />
69 <input type="submit" name="btnAddOptionValuePrice" class="plusList_
    btnInside" style="float:none;" value="Preis_hinzuf&uuml;gen" />
70 <br />
71 <br />
72 Priorit&auml;t: <input type="text" class="short" name="
    OptionPriority($tariffOptionValue.optionValue)" value="$!
    tariffOptionValue.priority" />
73 #break
74 #end
75 #end
76 </td>
77 #end
78 #set($counter = $counter +1)
79 </tr>
80 #end
81 <tr>
82 <td class="longLabelOptions"><br/>neuer Optionswert</td>
83 </tr>
84 <tr>
85 <td class="longLabelOptions">
86 #if(!$changeProductForm.possibleOptionValues || $changeProductForm.
    possibleOptionValues.isEmpty())
87 <input type="text" name="newOptionValue" id="newOptionValue" class="
    inpOption_length160"/>
88 <input type="submit" name="btnAddOptionValue" class="plusList_
    btnInside_floatRight" style="width:_20px;" value="" />
89 <br/>
90 #if($changeProductForm.oldOptionValues && !$changeProductForm.
    oldOptionValues.isEmpty())
91 <select class="selOption_length183" name="oldOptionValue" id="
    oldOptionValue">
92 <option value="" >neuer Optionswert</option>

```

```

93 <optgroup label="vorhandene_Optionswerte">
94 #foreach($oldOption in $changeProductForm.oldOptionValues)
95 <option value="$!stringTool.txt2xml($oldOption)" onclick="fillField
    ('oldOptionValue','newOptionValue');">$!stringTool.txt2xml(
    $oldOption)</option>
96 #end
97 </optgroup>
98 </select>
99 #end
100 #else
101 <select name="newOptionValue" class="medium" style="width:170px;">
102 #foreach($possibleOption in $changeProductForm.possibleOptionValues)
103 <option value="$possibleOption">$possibleOption</option>
104 #end
105 </select>
106 <input type="submit" name="btnAddOptionValue" class="plusList_
    btnInside_floatRight" style="width:_20px;" value="" />
107 #end
108
109 </td>
110 </tr>
111 ##end
112 <tr>
113 <td class="longLabelOptions">&nbsp;</td>
114 <td class="longInputOptions">&nbsp;</td>
115 </tr>
116 </table>
117 </div>

```

Listing A.4: Tarifimport mit Tarifooption

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tariffimport>
3   <tariff>
4     <tariffFormula>
5       <name>Power_OneMeter_Option</name>
6       <formula>result = AchievementPrice + (PowerConsumption *
          ConsumptionPrice);
7     </formula>
8   </tariffFormula>
9   <status>1</status>
10  <priority>0</priority>
11  <i18n>
12    <lang>
13      <locale>de_DE</locale>
14      <description>Oekostrom...</description>
15      <name>Oekostrom</name>
16      <agbFile />
17      <agbTemplate />
18      <vglFile />
19      <vglTemplate>agb/AVBEltV</vglTemplate>
20      <contractTemplate />
21    </lang>
22  </i18n>
23  <name>OEKOSTROM</name>
24  <signatureRequired>>false</signatureRequired>
25  <termExtension>1</termExtension>
26  <termExtensionUnit>2</termExtensionUnit>
27  <cancellationPeriod>1</cancellationPeriod>
28  <cancellationPeriodUnit>2</cancellationPeriodUnit>
29  <contractTerm>2</contractTerm>
30  <contractTermUnit>2</contractTermUnit>
31  <states>
32    <value>SN</value>
33  </states>
34  <counties />
35  <excludeCounties />
36  <postalCodes />
37  <excludePostalCodes />
38  <excludeCities />
39  <customerGroups>
40    <value>PRIVATE</value>
41    <value>BUSINESS</value>
42  </customerGroups>
43  <media>POWER</media>
44  <image>../images/stecker.jpg</image>

```

```

45 <tariffType>1</tariffType>
46 <tariffTemplate />
47 <tariffPart>
48   <name>PowerConsumption</name>
49   <i18n>
50     <lang>
51       <locale>de_DE</locale>
52       <displayName>Stromverbrauch</displayName>
53       <shortName />
54       <description />
55       <attributes />
56     </lang>
57   </i18n>
58   <prices />
59   <tariffPartType>2</tariffPartType>
60   <quantityUnit>Jahr</quantityUnit>
61   <quantityStep>1</quantityStep>
62   <minOrderQuantity>1</minOrderQuantity>
63 </tariffPart>
64 ...
65 <tariffPart>
66   <name>priceguarantee</name>
67   <i18n>
68     <lang>
69       <locale>de_DE</locale>
70       <displayName>Preisgarantie</displayName>
71       <shortName />
72       <description />
73       <attributes />
74     </lang>
75   </i18n>
76   <tariffPartType>3</tariffPartType>
77   <tariffOptionValues>
78     <tariffOptionValue>
79       <value>3 months</value>
80       <description>priceguarantee for 3 months</description>
81     <prices>
82       <price>
83         <modifyPart>ConsumptionPrice</modifyPart>
84         <currency>EUR</currency>
85         <value>0.00</value>
86       </price>
87     </prices>
88   </i18n>
89   <lang>
90     <locale>de_DE</locale>

```

```

91      <description>Preisgarantie f r 3 Monate</description>
92      <valueI18N>3 Monate</valueI18N>
93      </lang>
94    </i18n>
95  </tariffOptionValue>
96  <tariffOptionValue>
97    <value>6 months</value>
98    <description>priceguarantee for 6 months</description>
99    <prices>
100      <price>
101        <modifyPart>ConsumptionPrice</modifyPart>
102        <currency>EUR</currency>
103        <value>0.02</value>
104      </price>
105    </prices>
106    <i18n>
107      <lang>
108        <locale>de_DE</locale>
109        <description>Preisgarantie f r 6 Monate</description>
110        <valueI18N>6 Monate</valueI18N>
111      </lang>
112    </i18n>
113  </tariffOptionValue>
114  <tariffOptionValue>
115    <value>12 months</value>
116    <description>priceguarantee for 12 months</description>
117    <prices>
118      <price>
119        <modifyPart>ConsumptionPrice</modifyPart>
120        <currency>EUR</currency>
121        <value>0.03</value>
122      </price>
123    </prices>
124    <i18n>
125      <lang>
126        <locale>de_DE</locale>
127        <description>Preisgarantie f r 12 Monate</description>
128        <valueI18N>12 Monate</valueI18N>
129      </lang>
130    </i18n>
131  </tariffOptionValue>
132 </tariffOptionValues>
133 </tariffPart>
134 </tariff>
135 </tariffimport>

```

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Nentmannsdorf, 11. Januar 2010

Martin Patzig

